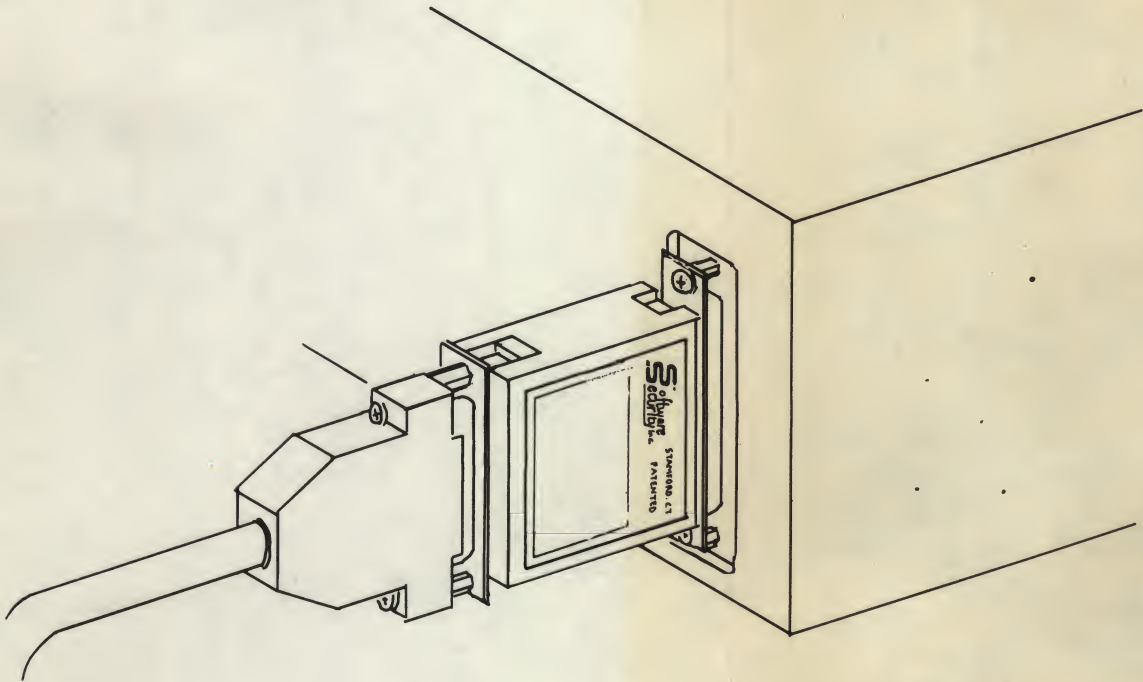


COPY PROTECTION

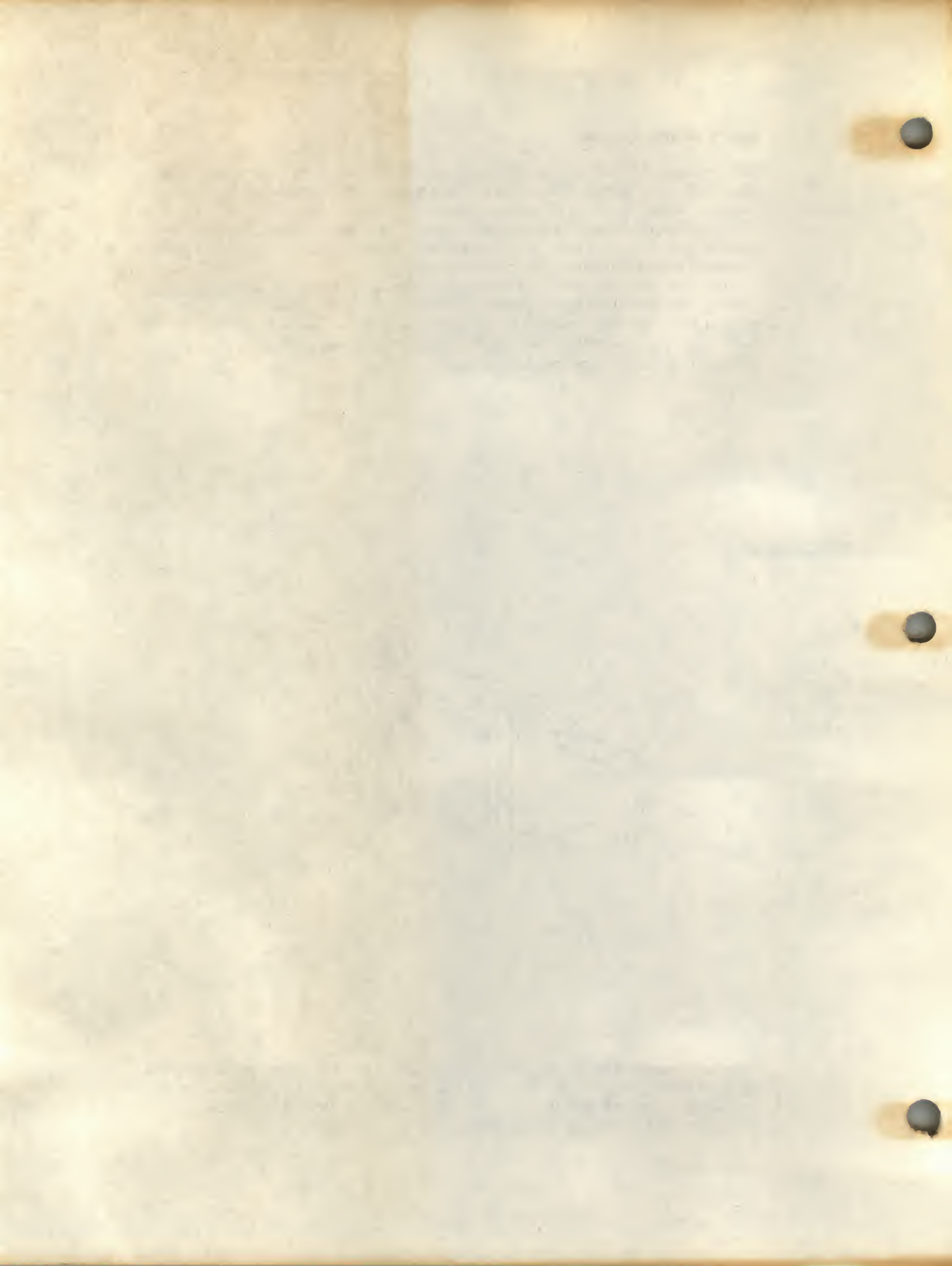
Earlier releases of ASYSTANT and ASYSTANT+ (prior to 1.04) were copy protected by using a "fingerprint" or Key diskette. That diskette has been replaced with a copy protect "block" which attaches to a parallel port of your computer, as described below.

In sections 1.1.1 through 1.1.3 of the documentation, references are made to the Key Diskette or the "fingerprint" for copy protection. Since the block is now being used, please ignore these statements and substitute references to the block instead.

The block must be installed on a parallel port of your computer. Usually, your computer will have only one parallel port, and your printer will be attached to it. The block can be placed between your parallel port connector and the printer connector without interfering with the printer operation, as illustrated in the figure below. Note that you may connect the block to any parallel port, if you have additional parallel ports.



To install the block on the parallel port used by your printer, first detach the printer cable, place the block on the port connector (it fits in only one way), screw the block to the port connector, and reattach the printer cable. You will now be able to enter ASYSTANT or ASYSTANT+ as described in section 1.1 of the documentation.



Dear ASYSTANT and ASYSTANT + Owner,

This is your Tutorial and Reference Manual for ASYSTANT(+) Ready-to-Run Scientific Software.

ASSEMBLY

Enclosed with this manual is a set of tab dividers. Please insert the dividers between the manual pages as follows:

TAB	LOCATION
Table of Contents	Before the Table of Contents
Tutorial	After the Introduction and before Session 1
Reference	After Session 12 / before Chapter 1
Appendices	After Chapter 21 / before Appendix A
Index	After Appendix E and before the Index

THE READ.ME FILE

A file named READ.ME is included on the ASYSTANT(+) System Diskette. READ.ME contains useful information on running ASYSTANT(+) with other resident software programs, such as macro facilities and disk space optimizers. READ.ME also offers some special tips for using ASYSTANT(+).

README is an ASCII file that can be reviewed using the ASYSTANT(+) text editor, the DOS TYPE command, or any standard word processor. We recommend that you print a copy of the README file and insert it in your ASYSTANT(+) manual. To print the file from DOS, follow these instructions:

1. Be sure your printer is turned on and online.
2. Insert the System Diskette in drive A.
3. At the A prompt (A>), enter:
A> COPY A:README PRN

Enjoy!

Asyst Software Technologies, Inc.

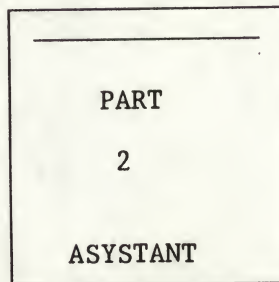
SOFTWARE PM2260

4822 872 89003

Dear user,

Enclosed you'll find an additional part of the operating manual.
This has to be put into the binder at the following places.

1. Part 1 has to be inserted at the beginning. This is before the actual ASYSTANT's manual starts.
2. Right before the part where the description of ASYSTANT starts, insert the page:



3. Tutorial session number 9 has to be inserted after tutorial session number 8 of the ASYSTANT manual (tutorial section).



PHILIPS

SECRET
- 123456

SECRET

PM 2260 OSCILLOSCOPE SIGNAL PROCESSING

The PM 2260 Oscilloscope Signal Processing (OSP) package combines specific instrument drivers for the PHILIPS Digital Storage Oscilloscope (DSO) family together with the ASYSTANT scientific software.

The oscilloscope drivers software allows measurement data, acquired by Philips Digital Storage Oscilloscopes via the GPIB, to be transferred directly to your PC. ASYSTANT can then be used to perform calculations on the transferred data.

Contents of the package:

- 1) 5 diskettes: 4 diskettes ASYSTANT.
1 diskette PHILIPS oscilloscope acquisition software.
- 2) 1 copy protection key.
- 3) ASYSTANT user manual.
- 4) PHILIPS data acquisition user manual.

Organization of the manual

The manual has been split up into 2 parts:

Part I describes the data acquisition part (Acquiring data using the GPIB bus).

Part II describes the processing software: ASYSTANT.

Part II starts with tutorial sessions, which are highly recommended in order to familiarize yourself with the PM2260 software package.

From these tutorial sessions, session 8 is not applicable for the PM2260 Oscilloscope Signal Processing Package and may be omitted.

Sessions 1 to 7 describe how to work with the ASYSTANT scientific software.

The last session: session 9 describes the acquisition of data with a PHILIPS Digital Storage Oscilloscope, and further processing of this data.

We strongly recommend you to go through session 9 as it describes the scope of the PM2260 software package.

MEMORANDUM FOR THE DIRECTOR, FBI

SUBJECT: [Illegible]

TO: [Illegible]

FROM: [Illegible]

DATE: [Illegible]

RE: [Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

PART

1

DATA-
ACQUISITION

1999

1999

1999

1999

1999

1999

1999

TABLE OF CONTENTS**PAGE****INTRODUCTION TO PM2260**

0.1 General Description	1
0.2 System requirements	1
0.3 The two meanings of acquisition	1
0.4 Notational conventions	1

GETTING STARTED

1.0 Hard-disk installation	2
1.1 Instructions for the owner of two floppy drive systems	3
1.2 Oscilloscope Signal Processing	3

THE ACQUISITION PROGRAM

2.0 General description	4
2.1 Menu selections	5
2.2 Menu layout	6
2.3 Color presentations	6
2.4 DSO local/remote state	7

MENUS

3.0 Introduction	8
3.1 Instrument menu	8
3.2 Register menu	10
3.3 Channel menu	11
3.4 Data points menu	12
3.5 Acquisition menu	14
3.6 Store menu	15
3.7 Quit menu	18
3.8 set Up menu	19
3.9 File handling	20

STATUS FIELD

4.0 Explanation	25
-----------------------	----

MACROS

5.0 Explanation	27
5.1 Example	27

SUPPORTED DSO'S

6.0 Supported DSO's	28
6.1 Tables of differences	29

FILE FORMATS

7.0 ASYSTANT format	31
7.1 ASCII format	31

Main body of handwritten text, consisting of several paragraphs. The text is extremely faded and illegible, appearing as light grey shapes against the cream-colored background. It seems to follow a standard letter format with an opening, a main message, and a closing.



INTRODUCTION TO PM2260

0.1 General Description

The use of ASYSTANT has now been extended by a fast and easy to use program which allows data communication with PHILIPS Digital Storage Oscilloscopes (DSO) using the GPIB. It acquires data and saves it to user defined files. These files can be read by ASYSTANT. Besides the register data, stored settings like timebase, attenuator etc, are read from the DSO. Thus creating the possibility of scaling data into the ASYSTANT environment.

The acquisition program also features data acquisition from ASCII files plus the possibility of sending the data back to the DSO. The program is menu driven and self-explanatory. User and other errors are reported on the display. Macros can be used to combine several actions into one keystroke.

0.2 System Requirements

Besides the minimum system requirements for ASYSTANT (see PART 2: tutorial session 1) the acquisition program requires the PM2201 GPIB interface board.

0.3 The two meanings of acquisition.

Throughout this manual the word "acquisition" has two meanings. For the PM2260 software package acquisition means transfer of data from DSO to PC using the GPIB. In the part of ASYSTANT it refers to the ASYSTANT+ program with acquisition via plug-in cards for the PC. This however is not applicable for the PM2260 package!

0.4 Notational conventions

Throughout this part of the manual the keys that need to be pressed are indicated by <key>. So <Enter> means press the enter or carriage return key rather than typing < E n t e r >.

GETTING STARTED

1.0 Hard-disk Installation

The first thing to do is to install the PM2260 software into a new subdirectory on your hard-disk. You can give this subdirectory any name, but the name **ASYSTANT** will be most suitable.

Type in the following DOS commands to make and enter a new subdirectory **ASYSTANT**.

```
MD \ASYSTANT <Enter>
CD \ASYSTANT <Enter>
```

The next step is to copy the ASYSTANT software into the new subdirectory. Place the first distribution diskette ('system diskette, white label ') into the A: drive, and type in the following DOS command:

```
COPY A:*. * <Enter>
```

Repeat this sequence for all the distribution diskettes in the PM2260 package. For more details on the contents of the ASYSTANT diskettes see tutorial session 1.1 **Getting Started** in PART 2 of this manual.

The following files must be copied from the acquisition diskette:

?????.EXE	The acquisition programs of supported DSO's. ???? = Digital Storage Oscilloscope type number.
OSP.BAT	Invocation program of the acquisition program and ASYSTANT.
DSO.SUP	Text file with type numbers of supported DSO's.

Before you can work with the acquisition program, the GPIB.COM, supplied with Philips software products must be installed.
Check the following:

A **CONFIG.SYS** file should be present in the root directory of your system. This file must contain the line
DEVICE = [path] GPIB.COM.

The file **GPIB.COM** should be present in the directory specified by [path].

The system must be reset after installation.

To start the PM2260 software package type in the following:

```
CD \ASYSTANT
```

```
OSP ???? <Enter> ( Replace ???? by the type number of your DSO. )
```

```
For example OSP 3320 <Enter>.
```

Continue with section 1.2

1.1 Instructions for the owner two floppy drive systems

We do not recommend the PM2260 software to be used on a dual disk drive system because of speed limitation and user unfriendly operation. For those who still want to try, act as follows:

Boot the system from a DOS system diskette (not the ASYSTANT system diskette) on which you should have installed the PM2201 GPIB software (see PM2201 manual).

Insert the acquisition diskette into the A: drive and type in the following:

OSP ???? <Enter> (Replace ???? by the type number of your DS0.)

For example OSP 3320 <Enter>

Acquire and save your data. But before you leave the acquisition program insert the ASYSTANT system diskette into the A: drive and the ASYSTANT key diskette into the B: drive. Before leaving the ASYSTANT program reinsert the acquisition diskette into drive A:.

1.2 Oscilloscope Signal Processing

OSP will start the acquisition program with the type number you entered. After having acquired and saved the data, leave the acquisition program by selecting the ASYSTANT option in the QUIT menu: the ASYSTANT program is automatically started.

With the ASYSTANT program you can enter the stored data and start processing. After leaving ASYSTANT the acquisition program is restarted and new acquisitions can be taken.

Return to DOS by selecting EXIT from the Quit menu.

THE ACQUISITION PROGRAM

2.0 General description

The acquisition program is menu driven and consists of eight main items. All eight items are the headers of eight pull-down menus and they are displayed horizontally on the first line of the screen.

Only one menu is displayed at the same time (in this case the Instrument menu). The most significant selections of the menus plus some additional information are displayed in the so called STATUS field (see section 4.0).

A permanent line of text is displayed at the bottom of the screen explaining which keys can be used.

A variable line of text is displayed on the one but last line of the screen explaining the action and the reason for the action to be taken.

= Instrument Register Channel Data_points Acquisition Store Quit set_Up

Search

STATUS

01-01-88		12:00:00	
Instrument : None			
Register 0	Channel A	Samples real	
Begin 0	End 4095	Count 1	
Data requested : 4096		Unscaled	
Data acquired : 0			

Press <Enter> to verify instruments connected to GPIB.

Move cursor with <↑><↓><←><→><Home><End><PgUp><PgDn>; Select with <Enter>.

2.1 Menu selections

There are two different ways to select menus.

2.1.1 Indirect

The following cursor keys should be used to select menus.

key	description	action
←	arrow left	Moves one menu to the left. If the current menu is the first menu the last menu is selected.
→	arrow right	Moves one menu to the right. If the current menu is the last menu the first menu is selected.
Home	Home	Selects the first menu.
End	End	Selects the last menu.

2.1.2 Direct

Type the capital letter from the menu header.

key	menu
I	Instrument
R	Register
C	Channel
D	Data points
A	Acquisition
S	Store
Q	Quit
U	set_Up

2.2 Menu layout

Each menu consists of one or more items.

A cursor (selected item is displayed in reverse video) points to the item to be selected.

To move the cursor up and down you can use the following cursor keys:

key	description	action
↑	arrow up	Moves cursor up one position. If the cursor is at the top of the list moves to the bottom of list.
↓	arrow down	Moves cursor down on position. If the cursor is at the bottom of the list it moves to the top of the list.
PgUp	Page Up	Moves cursor to top of list.
PgDn	Page Down	Moves cursor to bottom of list.

2.3 Color presentations

There are different types of items that can be selected. Move the cursor over the items and press <enter> or <+> to select the required item. Each item type is displayed differently.

item	appearance	description
process / submenu	yellow (+)	Another menu is called or an action takes place.
selected	✓ Highwhite (+)	Current selected item.
not Selected	Lightgreen (-)	Item that can be selected. Resets previous selected item.

Note! The (+) and (-) signs are the colors for monochrome displays.

(+) = High intensity.

(-) = Normal intensity.

The ✓ sign indicates current selections.

2.4 DSO local/remote state

It is the user's responsibility to set the DSO front panel settings (signal adjustment). During data transfer between the controller and the DSO, the DSO is in remote state and will not respond to any changes on the front panel (except for power off).

Once the data transfer has finished, the DSO is returned to local state; this means that the user can change the settings or make a new recording.

MENUS

3.0 Introduction

In this section all the selectable menus are described.

3.1 Instrument menu

= Instrument Register Channel Data_points Acquisition Store Quit set_Up

Search

This menu is only used during the first time you use the acquisition program (e.g. after installation).

The instrument menu consists of two parts:

3.1.1 Search

This procedure tests all GPIB addresses (1 through 30) to find if the entered DSO type number is present at one of the addresses (OSP ????). If you press <Enter> a message is displayed:

Connect Instrument(s) PM???? information instrument Press any key to continue
--

Before pressing any key make sure:

- To connect the instrument.
- That the instrument can be addressed as talker and listener.
- That the GPIB address is between 1 and 30.

After pressing any key the **Search** starts. A counter, displayed next to **Search**, keeps track of the GPIB addresses being checked. When the **Search** procedure is ready either the connected DSO's and their GPIB addresses will become selectable items in the **Instrument** menu, or if the entered instrument type number (????) has not been found, the following message occurs:

No instrument PM???? found on the GPIB bus Please do connect one Press any key to continue

3.1.2 Selectable DSO's

All the instruments found by the **Search** procedure are displayed sequentially in the **Instrument** menu in the following way:

PM???? #xx

PM???? stands for the DSO type number (Example: PM3320).

#xx indicates the GPIB address of the DSO (Example: #8).

≡ **Instrument** **Register** **Channel** **Data_points** **Acquisition** **Store** **Quit** **set_Up**

Search
✓ PM3320 #8

The first instrument found will automatically become the selected instrument (indicated by the ✓ sign).

Note! If only one DSO with the correct type number is found, the **Instrument** menu is cleared and the **Register** menu will be selected automatically. If two or more (maximum 4) DSO's are found, the **Instrument** menu remains the current menu.

3.2 Register menu

= Instrument Register Channel Data_points Acquisition Store Quit set_Up

✓ Register 0
Register 1

Reading data from a DSO can only be done from one of it's registers. Now one of the available registers has to be selected. The default selected register is Register 0. The number of registers depends upon the type of the oscilloscope.

See section 6 for an overview of supported Digital Storage Oscilloscopes and their differences.

Note! After selecting a register the Channel menu is automatically selected.

3.3 Channel menu

≡ Instrument Register **Channel** Data_points Acquisition Store Quit set_Up

✓ Channel A
Channel B
Channel ALL

This menu should be used if the DSO is in dual channel mode (A and B).
In case of single channel mode (A or B) the actual channel should be selected
for data transfer. The default selected channel is **Channel A**.

Note! After selecting a channel the **Data_points** menu is
automatically selected.

3.4 Data_points menu

≡ Instrument Register Channel **Data_points** Acquisition Store Quit set_Up

Begin	0
End	4095
Count	1
✓ Samples real	
Samples all	

The **Data_points** menu consists of two parts!

Part 1 specifying the amount of data that has to be acquired from the DSO.

Part 2 specifying the kind of samples that has to be acquired.

3.4.1 Amount of data

The size of a DSO register varies from instrument to instrument. In most cases the size will be 4K (4096). See also section 6.

A PHILIPS DSO is capable of sending parts of the register data. These parts are indicated by the **Begin**, **End** and **Count** factors. The **Begin** and **End** values are the limits and the **Count** value determines the amount of data points to be sent.

Example! Assume the following values.

Begin	0
End	9
Count	2

The data points, with the following sequence, are send: 0, 2, 4, 6 and 8 .
To change the values of the **Begin**, **End** or **Count** parameters type <Enter>.

The value is cleared from the screen and a small blinking cursor appears. Now a new value can be entered.

Use the following keys:

<0> .. <9>	Digits of new value.
<Enter> <↓>	Stores new entered value.
<Esc>	Ignores new entered value and restores old one.
<Backspace>	Deletes last typed digit.
<Ctrl><Backspace>	Deletes all typed digits.

≡ Instrument Register Channel **Data_points** Acquisition Store Quit set_Up

Begin	
End	4095
Count	1
✓ Samples real	
Samples all	

3.4.2 Samples

Depending upon the DSO timebase setting a register is filled with different amounts of measured samples (see DSO manual). PHILIPS DSO's can be asked to send only those measured samples (Samples real) or all the samples plus interpolated samples (Samples real).

The default values for the items are :

Begin	0
End	The maximum registers size minus one ($4K-1 = 4095$ bytes).
Count	1
Samples	Real

Note! After having entered a new value for any of the items, the cursor is moved down one position.
After selecting **Samples real** or **Samples all** the **Acquisition** menu is automatically selected.

3.5 Acquisition menu

= Instrument Register Channel Data_points Acquisition Store Quit set_Up

From instrument
From file

Two options can be selected from this menu. The acquisition of register data from the selected instrument or the acquisition of data from an ASCII file.

3.5.1 Acquisition from instrument

After pressing <Enter> the text **From instrument** will be replaced by the text **Reading data!** until the acquisition is finished. In the STATUS field on the screen the actual amount of data points, acquired from the DSO, is displayed (it might differ from the requested amount specified by the **Begin**, **End** and **Count** values due to the timebase setting of the DSO). Beside the samples also settings like attenuator, timebase etc, are read from the DSO. All data read from the instrument will remain in memory until a new acquisition is started or if the program is terminated.

3.5.2 Acquisition from file

The purpose of this function is to read data from an ASCII file so it can be sent to the DSO. For example you can generate a signal with ASYSTANT and save it to an ASCII file. The ASCII file can be read with this program and sent to the DSO (see section 3.6 Store menu). You might use the generated signal as a reference. See section 7 for more details on the format of the ASCII file.

After pressing <Enter> a new menu appears on the screen in which you specify the name of the file to read the data from (see section 4 file handling). after selecting **START READING**, this menu is cleared and the acquisition menu text **From file** is replaced by **Reading data!** until the acquisition is finished.

The values in the ASCII file must be in the range of the vertical resolution of the DSO registers. If for example a DSO has a 10 bits vertical resolution the values should be between -512 and +511.

Note! If the acquisition of the DSO register or ASCII file was successful the Store menu is automatically selected.

3.6 Store menu

= Instrument Register Channel Data_points Acquisition Store Quit set_Up

☒ Unscaled ☐ Scaled
Save to file Send to instrument

The main purpose of this menu is to save acquired data from the DSO to a file, or to send acquired data from a ASCII file to the DSO. It consists of 3 parts.

Part 1 specifies whether the acquired data should be saved scaled or unscaled.

Part 2 specifying the action to be taken; saving acquired data to a file or send it to the DSO.

Note! Unscaled and Scaled apply only to Save to file.

3.6.1 Scaling

Data acquired from the DSO can be saved Scaled or Unscaled.

Saving Unscaled will result in a file (ASYSTANT format) with 1 subfile, containing the acquired data.

Saving Scaled will result in a file (ASYSTANT format) with 2 subfiles.

Unscaled

If Unscaled is selected the pure register data will be saved. If for example your DSO has a 8 bits resolution (PM3350) the pure register data are integer values between 0 and 255 (From each value is 127 subtracted to get values between -127 and + 128). If the DSO has a vertical shift option, this option is taken into account. The vertical shift option is subtracted from all register data.

Scaled

Scaled data takes into account several stored front panel settings (timebase, attenuator). This option enables the user to perform calculations on scaled data. Data values are then calculated together with the stored front panel settings and saved into two subfiles.

Subfile 1 : X-axis related data.

In this subfile the number of data points calculated with the timebase settings are saved.

Subfile 2 : Y-axis related data.

In this subfile the data points calculated with the attenuator setting, probe factor and offset (if available for your DSO) are saved.

NOTE! If you read both channels (A and B) in one acquisition then there is a subfile for each channel.

Unscaled: Subfile 1: Channel A data
Subfile 2: Channel B data

Scaled : Subfile 1: X-axis related data
Subfile 2: Channel A Y-axis related data
Subfile 3: Channel B Y-axis related data

3.6.2 Save to file

After pressing <Enter> a new menu appears in which you can specify a file name to save the acquired data to (see section 3.9.1 how to enter a filename).

After selecting **START SAVING** this menu is cleared and the **Store** menu text **Save to file** is replaced by **Saving to file(s)** until finished. The file name, together with the number of subfiles that have been saved, is displayed in the **STATUS** field on the screen.

Note! If the saving of data to a file was successful the **Quit** menu is automatically selected.

3.6.3 Send to instrument

After pressing <Enter> the text **Send to instrument** is replaced by **Sending data!** until the transfer is complete. The data is sent to the register selected in the **Register** menu. The 'Dot join' on the DSO front panel is made active to get a clear picture in case less points than the full size of a register were sent.

If you send less points than the full register size from PC to the DSO, please note :

1. Make sure to clear the register manually (consult the DSO manual), otherwise the original signal will not be cleared.
2. The data points are displayed only over a part of the DSO screen.

If you send data from the PC to the DSO, front-panel settings do not correspond with the register data.

3.7 Quit menu

= Instrument Register Channel Data_points Acquisition Store Quit set_Up

ASYSTANT EXIT

Via the **Quit** menu you can leave this part of the PM2260 program. If you select **ASYSTANT** and the PM2260 package was started as described in section 1 the ASYSTANT program is started. Selecting **EXIT** will return the user to DOS. If **Auto save on** is selected from the **set Up** menu all current selections of this program are saved to a initialization file in the root called:

\OSP?????.INI (???? = the type number of the DSO)

So the next time this program is executed, the selections saved in this file are restored. If no such file is found, default settings are assumed.

Note! For dual floppy drive systems, the initialization file is saved to the floppy in the currently logged drive.

3.8 set_Up menu

= Instrument Register Channel Data_Points Acquisition Store Quit Set-up

✓ Color Monochrome
✓ Auto save on Auto save off

PM2260/3320 V1.1

With this menu you can select the monitor type and the auto save function on or off. Whenever you start this program **Auto save on** is selected. If you select **Auto save off**, no new initialization file is created. The old initialization file though is not erased and will be read the next time you start this program.

At the bottom of the **set_Up** menu the type and software version of the PM2260 can be examined.

Example : PM2260/3320 V1.1
 /3320 The type number of the DSO
 V1.1 The software release number of the PM2260

Note! (for monochrome displays only)
The default monitor type for this program is **Color**.
Therefore the cursor will not be visible on a monochrome display. Select the following key sequence for **Monochrome** display: <U> <↓> <Enter> .
After this the colors are adapted for monochrome displays.

3.9 File handling

This is the menu that appears after selecting **From file** from the **Acquisition** menu or after selecting **Save to file** from the **Store** menu. It consists of 2 parts.

Part 1 in which you enter the drive\path\filename separately (default drive\path\filename is assumed), or the matching filename for directory listings.

Part 2 in which you can activate data being read from, or saved to, a filename.

Filename	NONAME
Path C:	\ASYSTANT
Dir	*.ASY
START READING	

DIR
.
..
SQUARE1.ASY
SQUARE2.ASY
SINE1.ASY
SINE2.ASY
SINE3.ASY

There is one small difference between entering of this menu, depending upon which menu it was called from.

Called from **Acquisition** menu

- The text after **START** will be **READING**.

Called from **Store** menu

- The text after **START** will be **SAVING**.

You can leave this menu without any actions being taken by pressing <Esc> or <=>.

3.9.1 Filename

To change the filename type <Enter>.

Filename	
Path C:	\ASYSTANT
Dir	*.ASY
START READING	

The old name is cleared from the screen and a small blinking cursor indicates where to type the new name. All characters allowed for DOS filenames can be entered.

You can use the following keys:

<Enter> <↓>	Restores old filename.
<Esc>	Ignores new entered filename and restores old one.
<Backspace>	Deletes last typed character.
<Ctrl><Backspace>	Deletes all typed characters.

The maximum length of a filename is 8 characters.
The maximum length of an extension is 3 characters.
Separate the filename and extension with <.>.
Example of a filename 12345678.ABC

3.9.2 Path

If you want to save to or read from a file in another directory or disk, change the drive and or path name by using this menu. After pressing <Enter> a new menu appears below Path. Within this menu the full current path name is displayed. The blinking cursor is positioned at the end of the path name.

Filename	NONAME
Path C:	\ASYSTANT

C:\ASYSTANT_

All characters allowed for DOS filenames

You can use the following keys:

<Enter>	Stores new entered path. If the path name is longer then 8 characters only the last 8 character will be shown behind Path. ← will indicate that the path name is longer then 8 characters.
<Esc>	Ignores new entered path and restores old one.
<Backspace>	Deletes last typed character.
<Ctrl><Backspace>	Deletes all typed characters.
<Home>	Moves the cursor to the drive indicator. _ : After having typed a new drive name <A> _ _ <Z> the cursor is returned to the old position. To restore the old drive name press <Esc>.

Refer to the DOS manual for details on path names.

3.9.3 Dir

The contents of the directory, specified by the drive and path, is displayed in the field DIR.

DIR
.
..
SQUARE1.ASY
SQUARE2.ASY
SINE1.ASY
SINE2.ASY
SINE3.ASY
PROEF.ASY
TEST.ASY
↓ TEST1.ASY

This field can hold up to 10 filenames or subdirectory names. If more than 10 names are in the directory an indicates that there are more entries to follow. Which filenames that should be in the DIR field depends on the matching filename (Dir *.ASY).

To change this matching filename press <Enter>.

Filename	EXAMPLE.NUM
Path C:	\ASYSTANT
Dir	*.ASY_
START READING	

The matching filename is not cleared but a small cursor is positioned at the end of the matching filename. All characters allowed for DOS filenames can be entered.

You can use the following keys :

<*>	Wildcard character to match all filenames or extensions.
<?>	Wildcard character to match all characters.
<Enter>	Stores new entered matching filename.
<Esc>	Ignores new entered matching filename and restores the old one.
<Backspace>	Deletes last typed character.
<Ctrl><Backspace>	Deletes all typed characters.

After having entered a new matching filename the directory is searched for all matching names and subdirectory names. Those found are displayed in the DIR field. To scroll within the directory list in case there are more than 10 entries, press <Tab>. The cursor is now moved to the first name in the DIR field.

You can use the following keys to scroll the list.

<↓>	Moves the cursor down one position. If the cursor already reached the bottom of the field, the list is scrolled up one position.
<↑>	Moves the cursor up one position. If the cursor already reached the top of the field, the list is scrolled down one position.
<Home>	Moves the cursor to top of field and displays the first 10 names of the list.
<End>	Moves the cursor to bottom of field and displays the last 10 names of the list.
<PgUp>	Move the cursor to top of field and scrolls down 10 names of the list.
<PgDn>	Moves the cursor to bottom of field and scrolls up 10 names of the list.
<Tab>, <Esc>	Moves cursor back to the menu.

3.9.4 START READING / START SAVING

If you select **START SAVING** or **START READING** by pressing <Enter>, the menu and the DIR field are cleared from screen and the actual saving or reading takes place.

STATUS FIELD

4.0 Explanation

Because only one menu is displayed at the same time, all the significant selections plus some additional information are displayed in the so called **STATUS** field.

STATUS

01-07-88	09:44:40
Files saved: None	
Files read : None	
Instrument : None	
Register 0	Channel ALL Samples real
Begin 0	End 4095 Count 1
Data requested : 4096	Scaled
Data acquired : 0	

This field is permanently on the screen and it consists of three parts.

Part 1

01-07-88	09:44:40
Files saved: None	
Files read : None	

In this part the current date and time are displayed. Also after the program has saved any file, the name of the file and the number of subfiles saved, are displayed.

01-07-88	09:46:00
Files saved: SQUARE.ASY	2 subfiles
Files read : None	

Part 2

Instrument :	None		
Register 0	Channel ALL	Samples real	
Begin 0	End 4095	Count 1	

In this part the selected instrument, register, channel, sample type, begin, end and count values are displayed.

Part 3

Data requested :	4096	Scaled
Data acquired :	0	

In this part the number of data points calculated over the begin, end and count values are displayed behind the text **Data requested :**. The actual amount of data point acquired from the DSO is displayed behind the text **Data acquired :**. Because this value is in some cases not the same value as the data requested, a message with the reason why is displayed. The selection **Scaled / Unscaled** is displayed too.

Data requested :	4096	Scaled
Data acquired :	1024	due to TB 1 ms

MACROS

5.0 Explanation

Use macros in this program if several groups of actions need to be repeated. It saves a lot of time! On leaving this program the macros are saved to a file called:

\OSP?????.MAC (???? = the type number of the DS0)

The next time this program is started the macros are restored. 10 macros with a buffer of 50 keystrokes each are available under the function keys <F1>...<F10>.

To start editing a macro press <Alt> <function key>.

On the upper left corner of the screen the text F plus the function key number is displayed. This indicates that the program is editing a macro. From now on any key that is pressed is stored into the macro, until <Alt> <function key> is depressed or if the macro buffer is full.

To invoke an edited macro press only the function key. During the editing of a macro all function keys are ignored. Therefore it is not possible to call a macro within a macro.

Consider the following when using macros.

- Select menus in a direct way (see section 2.1).
- Be sure to use the right sequence of actions so no error might occur. Macros ignore errors!
- Use one macro to do the selections and the acquisition
Enter the filename manually.
Use another macro to start saving and to leave the program.

5.1 Example

Let's use function key <F1> to select **Register 1** and **channel B**.

Type the following keys:

<Alt><F1>	Starts editing macro F1.
<R>	Selects Register menu.
<↓>	Moves cursor over item Register 1 .
<Enter>	Selects Register 1 , and Channel menu.
<↓>	Moves cursor over item Channel B .
<Enter>	Selects Channel B , and Data_point menu.
<Alt><F1>	Closes macro F1.

The next time you want to select **Register 1** and **Channel B** just type <F1> instead of previous actions. It is the users responsibility to create macros for the right applications.

SUPPORTED DSO'S

6.0 Supported DSO's

This section shows an overview of the supported PHILIPS Digital Storage Oscilloscopes, together with the relevant hard- and software specifications. For more information, please consult the corresponding DSO operating manual.

Supported DSO's :

PM3320 (table 6.1)

PM3350 (table 6.2)

6.1 Tables of differences

table 6.1

Type: PM3320	
Registers	4 (0 .. 3)
Register size	4K (4096)
Vertical resolution	10 bits (-512 .. +511)
Horizontal resolution	see table 6.1.1
Sources	Channel A Channel B
Acquisition modes	1 channel (A or B) 2 channels (A and B) channels added (A + B) average MIN / MAX

table 6.1.1

Horizontal resolution			
channel mode	timebase / div	max resolution	
		real	all
single / added	5 ns .. 200 ns	512	4096
	200 ns .. 5 s	4096	4096
dual (both channels)	5 ns .. 200 ns	1024	4094
	200 ns .. 5 s	4094	4094

Table 6.2

Type: PM3350	
Registers	2 (0 .. 1)
Register size	4K (4096)
Vertical resolution	8 bits (0 .. 255)
Horizontal resolution	see table 6.2.1
Sources	Channel A Channel B
Acquisition modes	1 channel (A or B) 2 channels (A and B)

table 6.2.1

Horizontal resolution			
channel mode	timebase / div	max resolution	
		real	all
single	.5 us .. 2 ms 5 ms .. 50 s	512 4096	1024 4096
dual (both channels)	.5 us .. 2 ms 5 ms .. 50 s	1024 4096	2048 4096

FILE FORMATS

7.0 ASYSTANT format

Files saved by the acquisition program are in ASYSTANT format so they can be read directly by the ASYSTANT program. Data saved is of type REAL (single precision). The contents of a subfile can be examined in the comment field of the ASYSTANT file.

7.1 ASCII format

Files read by the acquisition program must be in ASCII format. The data values in the file must fit in the bit resolution of the DS0.

For example : 10 bits resolution: values between -512 and +511
 8 bits resolution: values between -127 and +128

ASCII files created by the ASYSTANT program can be read by the acquisition program.

Consider the following, if you try to create ASCII files yourself.

ASCII file layout:

[[value][separator] ...][EOF]

- value : May consist of the following digits "0123456789.Ee-+".
- separator : Any non value character.
- EOF : End Of File character (ASCII code 26).

PART

2

ASSYSTANT

TAN 9

2

PHOT 2 M 3 P



READY-TO-RUN SCIENTIFIC SOFTWARE

FOR DATA ANALYSIS

Asyst Software Technologies, Inc.
Rochester, New York

Asyst Software Technologies, Inc. does not warrant that the Program will meet the Customer's requirements or will operate in the combinations which may be selected for use by the Customer or that the operation of the Program will be uninterrupted or error free or that all Program defects will be corrected.

Asyst Software Technologies, Inc. does not and cannot warrant the performance or results that may be obtained by using the Program. Accordingly, the Program and its documentation are sold "AS IS" without warranty as to their performance, merchantability, or fitness for any particular purpose. The entire risk as to the results and performance of the Program is assumed by you.

Neither Asyst Software Technologies, Inc. nor anyone else who has been involved in the creation, production, or delivery of this Program shall be liable for any direct, incidental, or consequential damages, such as, but not limited to, loss of anticipated profits or benefits, resulting from the use of the Program or arising out of any breach of any warranty. Some states do not allow the exclusion or limitation of direct incidental or consequential damages, so the above may not apply to you.

IBM® is a registered trademark of International Business Machines Corporation.

Copyright © 1987 by Asyst Software Technologies, Inc.

All rights reserved. Reproduction or adaptation of any part of these programs or documentation beyond that permitted by Section 117 of the 1976 United States Copyright Act without permission of the Copyright owner is unlawful.

Asyst Software Technologies, Inc.

Rochester, New York

Printed in the U.S.A.

(ASYSTANT) ISBN: 0-02-879407-9

Designer: Betty Binns Graphics/Martin Lubin

TABLE OF CONTENTS

INTRODUCTION TO ASYSTANT

- 0.1 General Description ix
- 0.2 Minimum System Requirements x
- 0.3 Manual Organization xii
- 0.4 Technical Support xii

Section One: TUTORIAL

- SESSION 1: A Quick Look at the Desk Calculator T1-1
- SESSION 2: Arrays and Graphics T2-1
- SESSION 3: A Study of Pendulum Motion T3-1
- SESSION 4: Simulating the Motion of a Car Over Speed Bumps T4-1
- SESSION 5: Analysis and Processing of EKG Waveforms T5-1
- SESSION 6: A Statistical Analysis of Stock Portfolios T6-1
- SESSION 7: Diffraction Patterns and Two-dimensional Plots T7-1

Section Two: REFERENCE

- Chapter 1: General Features R1-1
 - 1.1 The ASYSTANT Environment R1-1
 - 1.2 System Configuration R1-6
 - 1.3 Using Help R1-14
 - 1.4 The NotePad Editor R1-16
 - 1.5 Using DOS Operations R1-23
 - 1.6 The Mini Calculator R1-24
 - 1.7 Error Handling R1-25
- Chapter 2: Desk Calculator R2-1
 - 2.1 Overview R2-1
 - 2.2 Entering Numbers R2-4
 - 2.3 Basic Arithmetic R2-6
 - 2.4 Arrays R2-8
 - 2.5 Editing the Command Line R2-11
 - 2.6 Calculator Functions R2-12
 - 2.7 The Array Operations Menu R2-17
 - 2.8 Conversions and Special Functions R2-23
 - 2.9 Wave and Matrix Operations R2-27
 - 2.10 The Command Line: an Alternative to Menus R2-33

Chapter 3:	User Functions	R3-1
3.1	Overview	R3-1
3.2	Function Editing	R3-3
Chapter 4:	Graphics	R4-1
4.1	Overview	R4-1
4.2	The Graphics Menu	R4-3
Chapter 5:	Waveform Generator	R5-1
5.1	Overview	R5-1
5.2	Main Menu Description	R5-5
5.3	Waveform Type Selection	R5-7
5.4	Horizontal and Vertical Specifications	R5-9
5.5	The Waveform Manipulation Menu	R5-11
Chapter 6:	Waveform Processor	R6-1
6.1	Overview	R6-1
6.2	The Waveform Processor Main Menu	R6-6
6.3	Manipulating the Segment and Waveform Repositories	R6-7
6.4	Setting the Plotting Options	R6-10
6.5	Selecting Vertical and Horizontal Axis Parameters	R6-11
6.6	Processing the Current Segment	R6-13
6.7	Analyzing the Current Segment	R6-15
6.8	Memory Options	R6-17
Chapter 7:	File Input/Output	R7-1
7.1	Overview	R7-1
7.2	ASYSTANT Data File Structure	R7-3
7.3	File/Var Transfer	R7-7
7.4	Converting File Types	R7-12
Chapter 8:	File Processor	R8-1
8.1	Overview	R8-1
8.2	Considerations in Selecting and Processing Data	R8-4
8.3	Input File Specification	R8-7
8.4	Output File Specification	R8-11
8.5	Processing Operations	R8-14
8.6	Selection of Input and Output Variables	R8-21
Chapter 9:	Curve Fitting	R9-1
9.1	Overview	R9-1
9.2	Fit Options	R9-5
9.3	User Function Fits	R9-11
Chapter 10:	Polynomial Operations	R10-1
10.1	Overview	R10-1
10.2	The Menu of Polynomial Operations	R10-2

Chapter 11:	Statistics	R11-1
11.1	Overview	R11-1
11.2	The Statistics Main Menu	R11-3
11.3	Hypothesis Testing	R11-6
11.4	Theoretical Distributions	R11-9
11.5	Analysis of Variance	R11-11
11.6	Stepwise Regression	R11-14

Chapter 12:	Differential Equations	R12-1
12.1	Overview	R12-1
12.2	Specifying the problem	R12-2

APPENDICES

APPENDIX A:	File I/O Routines	A-1
APPENDIX B:	Error Messages	B-1

INDEX

INTRODUCTION

0.1 General Description

The ASYSTANT is a powerful software tool specifically designed for scientists and engineers. Capable of carrying out the numerical and statistical analyses routinely required in almost all scientific pursuits, the ASYSTANT is as easy to use as a scientific calculator, and requires no programming.

ASYSTANT uses the full power of an IBM PC, PC/AT, HP Vectra, or other compatible to give you array processing at minicomputer-like speeds and interactive high-resolution graphics for data interpretation or presentation.

ASYSTANT+ has all of the capabilities of the ASYSTANT, plus an extensive set of options for acquiring data from laboratory instruments. ASYSTANT+ provides seven different, menu-driven acquisition modes which simulate traditional laboratory devices such as strip chart recorders, XY recorders, and signal averagers. It combines the precision and convenience of computerized data acquisition with the simplicity of conventional data acquisition methods. ASYSTANT+ supports all of the most popular data acquisition boards.

With ASYSTANT+ acquisition, data reduction, and analysis are combined in a single, comprehensive solution to many scientific and engineering problems.

Please note: Throughout this manual, the name ASYSTANT+ will be used when referring to the data acquisition options only available in ASYSTANT+. Where the name ASYSTANT is used, the information applies to both the ASYSTANT and ASYSTANT+ packages.

To make learning easier, ASYSTANT was designed to work like a calculator. But with ASYSTANT a single keystroke can perform a calculation on an entire array. The ASYSTANT also has extensive graphics capabilities for data presentation, and can produce graphics on either the display or an HP plotter. It supports all standard calculator functions, as well as a number of special operations such as Bessel and Gamma functions. It also supports matrix operations, polynomial operations, and complex numbers.

The system is menu driven, but numerical operations can also be executed through command lines. These can be assigned to function keys, allowing the user to automate repetitive calculations.

The ASYSTANT also features a built-in note pad editor available at any time to store notes about current computations on a disk file. The note pad has many features of a full scale text editor and produces files compatible with Word Perfect, WordStar, and most popular word processors. Numerical results generated by the ASYSTANT can be saved directly in the note file, or in data files, without retyping.

A wide range of additional capabilities enhance the basic calculator mode and make ASYSTANT even more powerful. These operations include:

- A *waveform generator* can create waveforms by summing wave trains of different shapes and different frequencies (including swept frequencies). It is far more flexible than the hardware waveform generator on which it is modeled. This feature offers capabilities for including different types of noise for simulation studies, graphical waveform editing, and a variety of other waveform manipulation options.

- A *waveform analyzer* lets you graphically select waveform segments for signal analysis and manipulation. Processing options include smoothing, spectrum analysis, and other data reduction operations, plus analysis of peaks (rise-time, peak width, areas), and many other operations. This menu option is modeled as a "waveform calculator", complete with storage cells for retaining intermediate waveform segments. It operates in an interactive mode that allows you to graphically "play" with waveforms, selecting segments and computing descriptive parameters on the fly.

- *File based processing* allows for conversion to and from ASCII and DIF files and easy transfer of data to and from ASYSTANT data files. It offers a wide variety of disk based computations including data smoothing, FFT's, convolutions, and signal averaging.

- An extensive *graphics* menu supports many different types of plots, including axonometric and contour plots. Plots can be directed to the screen or a pen plotter, to 9 different display windows, and can use up to 16 different colors with the IBM EGA graphics card. Graphic images can also be saved on disk, and then later recalled.

-- Solutions to systems of up to five *differential equations* can be numerically generated, and then automatically plotted.

-- *Curve fitting* can be performed for user defined functions, polynomials, exponential, logarithmic, and multilinear fits.

-- *Statistical analyses* can be calculated for tabular data, including basic statistics, histograms, and ANOVA tables. A menu of hypothesis testing is also provided, including both parametric and nonparametric tests. In addition, there is an interactive routine for performing stepwise regression.

0.2 Minimum System Requirements

The ASYSTANT requires the following hardware to operate:

- * An IBM PC, XT, or AT computer, HP Vectra, or equivalent 100% Compatible configured with a full 640 K of memory.

- * Two 360 K floppy disk drives, although a hard disk is highly recommended.

- * An 8087 or 80287 numeric coprocessor, depending on your computer. Check with your computer dealer for the correct chip.

- * One of the following graphics boards:

- IBM standard color/graphics adapter
 - IBM enhanced color/graphics adapter
 - Hercules monochrome graphics adapter
 - AT&T High Resolution
 - HP Multimode adapter
 - or a 100% compatible with one of the above.

- * HP7470 and 7475, or compatibles, is recommended (RS-232 only)

Additionally, ASYSTANT+ requires one of the following data acquisition boards:

- Cyborg Isaac 91-I Data Acquisition Systems
 - Data Translation DT2800 Series Boards
 - IBM Data Acquisition and Control Adapter (IBM-DACA)
 - Keithley Series 500 Systems (Selected Modules only)
 - MetraByte DASH-16
 - Tecmar LabMaster or Lab Tender
 - Dataq WFS-200PC

Check with Macmillan Software Company for current information on additional hardware support.

0.3 Manual Organization

The manual is divided into two sections:

Section 1: Tutorial introduces ASYSTANT through a series of brief example sessions. The tutorial is *strongly* recommended as a useful and easy means of learning to use the ASYSTANT.

Section 2: Program Reference contains detailed information about each feature in ASYSTANT. The first two chapters pertain to the system as a whole. Thereafter, a chapter is devoted to each Main Menu option and Acquisition Mode, with submenus treated in chapter subsections. Each chapter presents an overview of the feature under discussion and a detailed description of available options.

The best way to use this manual is to work through the Tutorial Sessions with your computer, consulting the Reference Chapters as needed. Once you are familiar with the ASYSTANT as a whole, it will be easy to locate technical information in the Reference covering the full range of ASYSTANT's capabilities.

0.4 Technical Support

The Macmillan Software Technical Support Hotline is available free of charge to new owners of ASYSTANT and ASYSTANT+ for 60 days from the date of purchase. Thereafter, users may enroll in our Extended Support Plan which includes ongoing Hotline support, free software maintenance upgrades, and discounts on new releases of ASYSTANT and ASYSTANT+. See the Customer Service Booklet enclosed with your diskettes for details.

SESSION 1: **A QUICK LOOK AT THE DESK CALCULATOR**

In this first session we will load and configure the ASYSTANT for your system and lead you on a brief tour through some features of the Desk Calculator. We will perform numerical calculations, set up a user defined function, enter some comments and results in the system's note pad, and use the on-line help facility.

The following functions will be used:

Desk Calculator

- to perform simple arithmetic,*
- to enter RPN and algebraic notation,*
- to store results in Variables,*
- to manipulate ASYSTANT variables as arrays (optional);*

User Functions

- to automate a sequence of calculator operations;*

Note Pad

- to store comments and preserve a record of calculations;*

DOS Operations

- to delete a file without leaving ASYSTANT;*

Help

- to look up instructions.*

1.1 Getting Started

The first thing that you should do after unwrapping your copy of ASYSTANT is to make sure that the distribution diskettes contain the right files. ASYSTANT owners should receive four diskettes, ASYSTANT+ owners six. Place them in a floppy disk A:drive to verify their contents. List their directories using the DOS command:

```
DIR A:
```

The System Diskette (white label) should contain the files:

ASYSTANT.COM	-- System COM file
CFG.SYS	-- System Overlay file (required at boot)
ASYSTANT.CFG	-- System configuration file
PRINTER.TBL	-- Printer Driver file
ASYSTANT.MSG	-- ASYSTANT error messages (duplicated on the Analysis diskette and, for ASYSTANT+, on the Acquire 1 and 2 diskettes).
ACQUIRE@.CFG	-- ASYSTANT+ Configuration File (<u>ASYSTANT+ only</u>)

The Key Diskette (red label) should contain the files:

ASYSTANT.OVL	-- System OVL file
EKGDMO.DAT	-- Sample Data file for tutorial example

The Help diskette (green label) should contain the file:

ASYSTANT.HLP	-- ASYSTANT help text
--------------	-----------------------

The Analysis Diskette (yellow label) should contain the overlay files for the ASYSTANT Main Menu Options:

CURVFIT.OVR	-- Curve Fit overlay
DIFFEQS.OVR	-- Differential Equations overlay
FILEPROC.OVR	-- File Processor overlay
FILEIO.OVR	-- File Input/Output overlay
GRAPHICS.OVR	-- Graphics overlay
POLYOPS.OVR	-- Polynomial Operations overlay
STATOPS1.OVR	-- Statistics overlay, part 1
STATOPS2.OVR	-- Statistics overlay, part 2
WAVGEN.OVR	-- Wave generator overlay
WAVPROC.OVR	-- Wave Processor overlay
ASYSTANT.MSG	-- ASYSTANT error messages (duplicate file)

If you purchased ASYSTANT+ you should receive two data acquisition diskettes, Acquire 1 and Acquire 2.

If you purchased ASYSTANT+ you should receive two data acquisition diskettes, **Acquire 1** and **Acquire 2**.

Acquire 1 (blue label) containing the files:

SC.OVR	-- Strip Chart Recorder overlay
HWS.OVR	-- Hardware Scroller overlay
XY.OVR	-- XY Recorder overlay
TR.OVR	-- Transient Recorder overlay
DLPE.OVR	-- Data Logger overlay, part 2
DLAQ.OVR	-- Data Logger overlay, part 1
TREKG.DAT	-- Data file for tutorial example
TREKG.DFP	-- Setup file for tutorial example
HSEKG.DAT	-- Data file for tutorial example
HSEKG.DFP	-- Setup file for tutorial example
ASYSTANT.MSG	-- ASYSTANT error messages (duplicate file)

Acquire 2 (violet label) containing the files:

HS.OVR	-- High Speed Recorder overlay
SA.OVR	-- Signal Averager overlay
FG.OVR	-- Function Generator overlay
CFGACQ.OVR	-- Configure overlay
CONVERT.OVR	-- Conversion overlay
THRMCP.LNBS	-- Thermocouple linearization database
ASYSTANT.MSG	-- ASYSTANT error messages (duplicate file)

**** If you are missing a diskette, or if files are incomplete or damaged, contact your dealer or Macmillan Software directly ****

1.1.1 INSTRUCTIONS FOR HARD DISK OWNERS

Hard disk owners will find it easiest to copy all files to a separate subdirectory on the hard disk. If you are not familiar with using subdirectories, consult your DOS manual. Place the first distribution diskette in the A: drive, and type in the following DOS commands:

```
C> md \asystant
C> cd \asystant
C> copy a:*. * /v
```

"md" and "cd" are DOS commands for "make directory" and "change directory". The "/v" parameter in the DOS "copy" command is used to verify the accuracy of the copy. Refer to your DOS manual for details if necessary.

Please note: for the balance of this Session, and in all subsequent Sessions, we adopt the convention of printing commands to be typed into your computer in **boldface**. Commands which need not be entered but which are discussed in the text are contained in quotation marks (" "). Don't include the quotation marks if you type the command at your keyboard.

When the first diskette has been copied, insert the second diskette and hit the F3 key to repeat the "a:*. * /v" command. (If you make a mistake just type the command in again). Keep going until all of the diskettes are copied (you don't need to copy the backup diskette as it is identical to the original).

Now you are ready to load ASYSTANT. Remember, however, that when you restart your computer or if you change directories, you must first type "cd \asystant" in DOS with a C> prompt before you can use ASYSTANT.

You will need to keep one copy of the Key Diskette available to boot the system. Lock away the backup copy and the originals of all other diskettes. *You must have a copy of a distribution Key Diskette in a floppy drive when you boot the system.* The ASYSTANT will not load unless it finds this diskette. After loading, you are free to remove it at any time. (Please continue with section 1.1.3).

1.1.2 INSTRUCTIONS FOR OWNERS OF TWO FLOPPY SYSTEMS

We recommend that if you intend to make heavy use of ASYSTANT (and we hope you do), you consider investing in a hard disk. Besides the convenience of not having to swap floppies in and out of drives, a hard disk system will run ASYSTANT more quickly (especially as you move between menus), and provides higher throughput rates in some data acquisition modes.

This said, using ASYSTANT on a two floppy system is straight-forward. We *strongly* recommend that you make copies of all of the distribution diskettes except the Key Diskette, and use the back up copies, not the originals! The Key diskette contains a copy protected fingerprint so you will have to use it. This is why a backup Key diskette is provided. Lock away the backup copy of the Key Diskette, and all distribution copies of the other diskettes. Keep them as backups in case something happens to the original Key diskette or the copies you just made.

WARNING

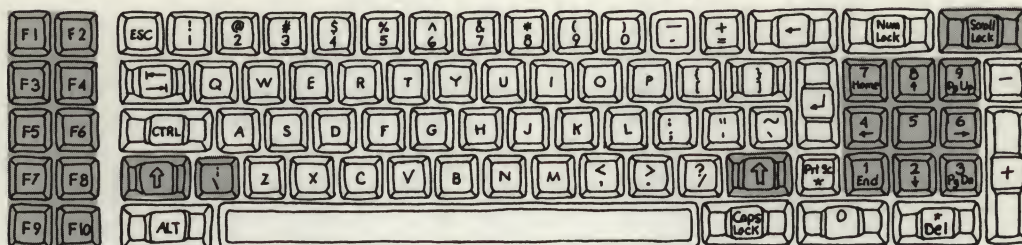
Do not place write-protect tabs on your working copies of the ASYSTANT diskettes. ASYSTANT will need to write to these disks.

You should also have a blank formatted diskette available: ASYSTANT will use this for temporary work files. (If you don't know how to format a diskette consult your DOS manual under the Format command). Label this floppy "ASYSTANT Temporary Work Files".

Now to get started, place the System Diskette in the A: drive, and log onto that drive (a: <Enter>). Next place the Key Diskette in the B: drive.

1.1.3 FOR OWNERS OF BOTH HARD DISK AND TWO FLOPPY SYSTEMS

Please note: The ASYSTANT may be loaded and executed from files that are copied from the original Key Diskette (eg., on your hard drive), but the original diskette must be in one of the system floppy drives in order for the program to boot, even for hard disk owners. The Key diskette contains a special copy protection "fingerprint" that cannot be copied to new diskettes, or to a hard disk. The Key diskette must *not* be write protected (i.e., do not place a write-protect tab on the Key diskette). Never write data or copy files to the Key Diskette. In some cases it may erase the fingerprint, making the diskette unusable. If anything happens to your Key diskette so that you are unable to use the ASYSTANT you can return it to Macmillan Software for a replacement (see the Customer Service Booklet enclosed with your diskette envelope).



<u>Key</u>	<u>Abbreviation</u>	<u>Key</u>	<u>Abbreviation</u>
Function Keys	F1-F10 or Fn	Keypad:	Cursor control keys
Escape	Esc	Home	Home
Tab	Tab	Up arrow	Up arrow
Control	Ctrl	Page Up	PgUp
Shift	Shft	Left arrow	Left arrow
Backslash	\	Right arrow	Right arrow
Alternate	Alt or @	End	End
Space Bar	Space	Down arrow	Down arrow
Enter	Enter or <CR>	Page Down	PgDn
Number Lock	Num Lock	Insert	Ins
Break	Break	Delete	Del

TABLE 1.1

*Key Name Abbreviations
(for Standard IBM PC Keyboard)*

We're now ready to start. Type **ASYSTANT**. First a banner and copyright notice will appear, and after a few seconds the prompt:

```
Recall/Setup/Continue *C* _
```

```
Recall functions, variables, and parameters from a disk file
```

```
Setup the hardware configuration
```

```
Continue
```

should come up on the display. You have now entered the **ASYSTANT**! As soon as you see this prompt, remove the Key Diskette from the drive, slide it into its protective envelope, and store it in a safe place. Dust, spilled coffee, and heavy objects are common ways to ruin the Key diskette so treat it carefully. If you are running a two floppy system, insert a blank, formatted diskette in the B: drive. **ASYSTANT** will use this as a scratch diskette to hold temporary work files.

The prompt **"*C*"** on the screen tells you two things. First, "C" for continue is the default setting. You could select Continue simply by hitting "Enter" -- however don't do that now. (See Table 1.1 for an explanation of the names used for keys in this manual). Second, the asterisks signify that this is an "action prompt". That is, **ASYSTANT** will act when you enter a value in an action prompt. "R" calls a special menu to recall a setup file; "S" calls a different menu to change the hardware configuration, and "C" puts you into the **ASYSTANT** itself.

Since this is your first time running **ASYSTANT**, it is probably necessary to set up **ASYSTANT** for your PC's hardware configuration. Type S in response to the action prompt on the display. **ASYSTANT** will display a menu of configuration options in the upper left hand corner [refer to Screen 1.1]. To the right of each line is the current setting for each menu option.

Configure	Current
Display	IBM Color Graphics Card, Black/White Monitor
Plotter	No plotter configured
Printer	IBM Graphics Printer
Files	System Overlay Files on drive C:
Return	Temporary Work Files on drive C:
	ASYSTANT Help File on drive C:

SCREEN 1.1

Note that the word "Display" is highlighted on the menu. If you hit the **Down Arrow** key, the highlight will move down the list. (If the prompt does not move, and the system beeps at you, it is probably because the key labeled "Num Lock" is incorrectly set. Hit "Num Lock" once to toggle it to the correct position). You can skip to the bottom item ("Return") by hitting the key labeled **End** and to the top item by hitting **Home**. All ASYSTANT menus use these keys in the same way.

To make a choice, you must move the highlight to the desired item and press "Enter". In this case, highlight **Display** then hit **Enter** [refer to Screen 1.2]. The lower window will show a menu of options for display hardware, and a cursor will be blinking to the right of the **Display Hardware (1-10):** prompt.

Configure	Current
Display	IBM Color Graphics Card, Black/White Monitor
Plotter	No plotter configured
Printer	IBM Graphics Printer
Files	System Overlay Files on drive C:
Return	Temporary Work Files on drive C: ASYSTANT Help File on drive C:

Display Hardware Types	Press Esc to return

1 -- EGA Card with more than 64k, and an Enhanced Color Monitor	
2 -- EGA Card with more than 64k, and a Standard Color Monitor	
3 -- EGA card with 64k, and an Enhanced Color Monitor	
4 -- EGA card with 64k, and a Standard Color Monitor	
5 -- IBM Color Graphics Card, and a Standard Color Monitor	
6 -- IBM Color Graphics Card, and a Black/White Monitor	
7 -- Hercules Graphics Card, and a Black/White Monitor	
8 -- AT&T High Resolution Graphics Card, and monitor	
9 -- HP Multimode Graphics Card, and monitor	
10 -- EGA Card, and a Monochrome monitor	
Display Hardware (1-10): < 6 >	Color on: < N >

SCREEN 1.2

1.1.4 PROMPT LISTS AND THE DISPLAY OPTION

Press **Enter** without typing in any number. Note that the cursor will skip to the **Color on: <N>** prompt. Hit **Enter** again and the cursor will skip back to its original position. Note you can also use the right and left arrow keys to move the cursor. These prompts make up a "prompt list". Prompt lists are used throughout the ASYSTANT for entering data. To change one of the parameters, simply position the cursor beside the current value, and type in a new value. Prompts which require several characters will not change until you press the "Enter" key. For example, the "Display Hardware:" prompt can be two characters "10"), so it requires you to press **Enter**. Prompts which require only a single character -- the "Color on:" prompt and all action prompts -- will accept your keystroke instantly, provided it is a valid response. ("Color on:" will only accept "N" unless the Display Hardware mode is set to 1, 2, 3, or 4).

Note that the two prompts show the default current settings surrounded by the <> characters. <> are used to show that this is an input prompt, not an action prompt. That is, input to these settings will not be acted on until "Esc" is pushed or an action prompt is selected. Throughout ASYSTANT, "Esc" has the effect of accepting the new prompt list values. Until you hit "Esc" you are always free to return to an input prompt and change the current setting without immediately affecting system operation.

Now let's select the proper settings for your hardware. EGA and CGA are IBM terminology. EGA stands for Enhanced Graphics Adapter; CGA stands for the older Color/Graphics Adapter. Most compatible equipment emulates the IBM CGA; check the documentation for your computer. If you are still unsure, call your dealer or the manufacturer of your system. ASYSTANT also supports the HP Multimode Card, the High Resolution mode on the AT&T 6300 series, and the Hercules Graphics Card. You need to enter the number corresponding to your display hardware in the "Display Hardware" prompt. Remember to hit Enter.

The "Color on:" prompt works only if you have selected one of the EGA options. Enter N for "No" if you intend to do screen dumps to a graphics printer: color can interfere with proper printing. Otherwise, select Y because the displays are greatly enhanced by using color.

When you have entered the correct entries, hit Esc. The display hardware menu will disappear, and the "Configure" Menu will again be active. If you changed the display setting, the new selection should now show up in the "Current" window.

1.1.5 THE PLOTTER OPTION

If you have an HP7440, 7470, 7475 or compatible plotter (such as a Gould Colorwriter) and wish to use it during a session, you will want to select the Plotter option. Press the **Down Arrow** to move the highlight to "Plotter" and hit **Enter** [refer to Screen 1.3]. Select the number corresponding to your plotter and then hit **Esc**.

WARNING

To use a plotter you must execute the proper MODE command from DOS to initialize the serial port before entering the ASYSTANT. Reference section 1.2.2 describes proper installation of the plotter and the mode command.

A plotter is not required for any examples in this tutorial.

Configure	Current
Display	IBM Color Graphics Card, Black/White Monitor
Plotter	No plotter configured
Printer	IBM Graphics Printer
Files	System Overlay Files on drive C:
Return	Temporary Work Files on drive C: ASYSTANT Help File on drive C:

Press Esc to return

Plotter Types

0 -- None
1 -- HP7440
2 -- HP7470
3 -- HP7475

Type of Plotter (0-3) < 0 >

SCREEN 1.3

1.1.6 THE PRINTER OPTION

ASYSTANT will support the graphics capabilities and other special characteristics of several popular printers. Select **Printer** (remember to hit **Enter**) and select your printer from the menu. If your printer is not on the menu, call the Macmillan Software Technical Hotline. Customized support for your particular printer may be available. The top menu options are reserved for custom support.

1.1.7 THE FILE OPTION

ASYSTANT comes configured for hard disk users, with all files designated on the C: drive. Two floppy system owners must select the File option and change all three prompts before proceeding [refer to Screen 1.4]. We recommend you select the A: drive for System Overlay files. This means that once ASYSTANT is configured (by selecting "Return" in the menu of Setup options) you can remove the System diskette from drive A: and replace it with one of the overlay diskettes: the Analysis diskette or, for ASYSTANT+, one of the Acquire diskettes. Each overlay file is associated with one of the ASYSTANT "Main Menu" options which are discussed later in this session. When you select a Main Menu option, its overlay must be present in the System Overlay drive.

Configure	Current
Display	IBM Color Graphics Card, Black/White Monitor
Plotter	No plotter configured
Printer	IBM Graphics Printer
Files	System Overlay Files on drive C:
Return	Temporary Work Files on drive C: ASYSTANT Help File on drive C:

Press Esc to return

File Drive Specifications

Disk Drive for System Overlay Files (A-Z): < C >
 Disk Drive for Temporary Work Files (A-Z): < C >
 Disk Drive for Asystant Help File (A-Z): < C >

SCREEN 1.4

We suggest you specify the B: drive for Temporary Work files. If you've been following instructions exactly to this point, the diskette labeled "ASYSTANT Temporary Work Files" should already be in the B: drive. The ASYSTANT needs up to 128K of space on this disk to ensure proper operation. Thus using the Key or System diskettes as a scratch diskette may result in an error in some system configurations, and also causes unnecessary wear.

We suggest you also specify the B: drive for the ASYSTANT Help file. When you use the online help system, you will be prompted to insert the Help diskette in the B: drive.

Press Esc to accept the values you have entered.

1.1.8 THE RETURN OPTION

You now should be ready to return to the initial prompt screen. Move the highlight to the **Return** item and press **Enter**. You will note that the disk drive light will go on momentarily. ASYSTANT has written the information you just specified in the **ASYSTANT.CFG** file. From now on -- unless you change them again -- these settings will be used as defaults in subsequent sessions. You will not need to take this configuration step again unless your hardware changes. Note that the configuration file was written on the same disk as the **ASYSTANT.COM** file, and should not be moved to another drive.

1.2 Desk Calculator Organization

Now we are ready to start working with the ASYSTANT. If you are working on a floppy based system, remove the System (white) diskette from drive A and replace it with the Analysis (yellow) diskette. When you load and configure ASYSTANT the System disk must be present. Thereafter it is not used by ASYSTANT. Instead you can insert one of the overlay disks: Analysis to select any Main Menu option (discussed below) and, for ASYSTANT+, the Acquire 1 diskette to select any of the first five data acquisition options and Acquire 2 to select from the last five. ASYSTANT+ data acquisition options will be covered in session 13.

TIP (for ASYSTANT+ owners on floppy based machines)

To limit the amount of diskette-swapping needed to run ASYSTANT+, you may wish to make your own custom overlay disks, rather than simply creating backup copies of the Analysis disk and the Acquire disks 1 and 2. Each overlay file on these disks is specific to its associated menu option. Thus, options which you do not frequently use do not need to be on your overlay disk.

Copy the overlays for whichever options you expect to frequently use from the Analysis and Acquire diskettes onto one disk and be sure to include the **ASYSTANT.MSG** file as well. You can then keep this disk in your system overlay drive (usually drive A) once ASYSTANT+ is booted and you will not need to swap disks as you move from the Desk Calculator to the Acquire options, etc.

Press **C** to continue on to the Desk Calculator. Whenever you enter the ASYSTANT you enter the Desk Calculator first. [Refer to Screen 1.5]. It is a rather complex display, containing 5 separate windows. We explain each below:

Stack Contents					Parameters				
<empty stack>					A= .0000				
					B= .0000				
					C= .0000				
					D= .0000				
					E= .0000				
					F= .0000				
					G= .0000				
					H= .0000				
					I= .0000				
Calculator Functions					Variables				
next	store	pedit	status	print	R= .0000				
dup	drop	swap	roll	pi	S= .0000				
sin	asin	sinh	asinh	inv	T= .0000				
cos	acos	cosh	acosh	neg	U= .0000				
tan	atan	tanh	atanh	abs	V= .0000				
exp	ln	10exp	log	sqrt	W= .0000				
Main Menu Options					X= .0000				
Acquire	Graphics	Functions			Y= .0000				
File Proc	File I/O	Polys			Z= .0000				
Wave Proc	Curve Fit	Diff Eqs							
Wave Gen	Stats	Save/Exit							

SCREEN 1.5

Stack Contents -- Displays the current contents of the system stack. The stack is used to hold numbers and arrays that are actively being worked on. Think of the stack as five registers used by the calculator for temporary storage during calculations. Arithmetic operations and functions all act on the argument(s) on the top of the stack.

Calculator Functions -- Contains a menu of basic calculator functions. This menu can be interchanged with 3 supplementary menus: "Array Operations", "Conversions and Special Functions," and "Wave and Matrix Operations".

Main Menu -- This menu provides access to Waveform Processing, Graphics, Disk I/O, and other ASYSTANT operations.

Parameters -- Displays the contents of the nine system Parameters A-I. These Parameters are used to store numbers of any data type and are analogous to calculator memories. They cannot contain arrays.

Variables -- Displays the contents of the nine system Variables, R-Z. Variables are memories, like Parameters, but more powerful: they can hold arrays as well as numbers.

The Main Menu is active when you first enter the Calculator. To switch back and forth between the calculator functions and the Main Menu, use the "PgUp" and "PgDn" keys. In this first session we will spend most of the time investigating the Desk Calculator, so press **PgUp**. Notice that the highlighted item in the Main Menu has now been turned off, and instead the "next" item in the calculator menu is turned on. The highlight indicates the menu item that you are currently poised to execute. It can be moved through the menu by using the arrow keys, just as you did in the "Setup" Menu. The difference here is that you can move both up and down and sideways. To actually execute the current menu item, you press the "Enter" key.

Note: Throughout this tutorial, we will ask you to "select" or "execute" a menu item. By convention, this will mean to move the highlight to an item and press "Enter".

1.3 ASYSTANT Arithmetic

Now let's try some simple arithmetic. We first need to enter some numbers. Try typing the digits 5.0. As soon as the first digit is pressed, the Main Menu window is repainted with the prompt:

Command: 5

You should type the decimal point, and terminate the number with the Enter key. Continue with the number 12.3. The first key you press will bring up the command line in the lower left window. It will stay up until you press Enter. Each time you enter a number it is placed on the top of the stack and the previously entered numbers are pushed down to make room for it. This is known informally as a "push-down" stack, or formally as a last-in/first-out (LIFO) stack. If you have used a Hewlett-Packard calculator, you have experienced them before.

The stack should now contain:

```
12.3000 (REAL)
5.0000 (REAL)
```

The number 12.3 is now at the top of the stack, and 5.0 has been pushed down a notch. Now enter the number 2.7. The stack contents become:

```
2.7000 (REAL)
12.3000 (REAL)
5.0000 (REAL)
```


Next we will add together the two numbers on the top of the stack. Press the + key, and Enter. ASYSTANT "pops" the top two numbers off the stack, adds them together, and pushes the result back on the stack. The stack contents are now:

```
15.0000 (REAL)
 5.0000 (REAL)
```

Suppose that we then want to divide the 15.0000 by the 5.0000. The "/" key will perform this division, but the arguments must first be arranged in the proper order. The "/" operation divides the second stack entry by the top entry, and leaves the result in place of the two original arguments; so before we can use / we must first interchange the 5.0000 and the 15.0000. The "swap" option in the calculator menu does precisely that. It switches the first and second stack entries. Use the arrow keys to highlight swap, and then press Enter. The 5.0000 and 15.0000 will then be swapped, with 5.0000 on top. Now press / Enter and you will get the correct result:

```
3.0000 (REAL)
```

The command mode can be used to carry out this entire calculation in one fell swoop. We have already seen the command mode; it is invoked whenever a number or math operator is pressed. It is also invoked by pressing any letter, "Esc", and "\", among other keys. Several numbers can be typed in at once *provided that they are separated from each other by at least one space*. Furthermore, arithmetic operations and menu options can be included in the command line, again provided that they are separated from adjacent commands by a space. Thus, the above calculation can be performed in a single command line:

```
Command: 5.0 12.3 2.7 + swap /
```

The syntax being used here is called Reverse Polish Notation (RPN). In RPN, arguments precede operations rather than being intermixed with them, so instead of $1 + 2$ you would write $1\ 2\ +$. This syntax is very efficient. It saves keystrokes because arguments for functions and operations don't have to be explicitly specified, they are automatically taken from the top of the stack.

If you are new to RPN, you may find it easier to take operations one step at a time, for example: $12.3\ 2.7\ +\ 5.0\ /$.

Although ASYSTANT normally works in RPN, it also allows you to *directly evaluate algebraic expressions* by entering them on command lines. To indicate that a command line is to be interpreted algebraically you must enter a backslash, "\", as the *first* character in any line. Let's try the same calculation as before, this time using algebraic notation:

```
Command: \ (2.7+12.3)/5.0
```

Notice that the command line appeared as soon as you pressed the "\". Also note that spaces are not necessary between numbers and operations when

using algebraic syntax. In the beginning it may seem most natural to stick with the algebraic syntax, but before long you will find yourself using a combination of algebraic and RPN. Remember, however, not to mix RPN and algebraic notation in any single command line and that the "\" must be the first character in a line when using algebraic notation. You will also find it useful to combine menu operations with command lines. This "pidgen" form is often easiest to learn and use, although it's harder to explain.

Please note: ASYSTANT command words such as "swap" can only be combined with other functions in RPN, or else they must be entered alone on the command line. They cannot be included in algebraic expressions.

Try doing the following problems in different ways to get used to the way the system works.

$$(\text{Sin}(-2.3+3.5))^2$$

$$\text{Log}(15/2.7)+\text{Log}(10/1.2)$$

$$(4.32-1.98)/(16.4-8.6)$$

In both RPN and algebraic modes, the standard arithmetic operators are entered by pressing the following keys:

Function	Key
Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponentiation	^

In RPN, execution flows from left to right in any expression. In algebraic expressions, operations are evaluated in the following order:

1. Expressions within parentheses
2. Exponentiation
3. Multiplication, division
4. Addition, subtraction

Operations of equal precedence are evaluated from left to right. Remember, operator precedence is meaningful only for algebraic expressions. RPN operates in a strict left-to-right order.

1.4 Using Parameters and Variables

The ASYSTANT provides eighteen Parameters and Variables for storing numbers and arrays. The Parameters are labeled A, B, C, D, E, F, G, H and I. They are reserved for single numbers and cannot be used to store arrays. The Variables are labeled R, S, T, U, V, W, X, Y and Z, and can store either numbers or arrays. For now we will only be using numbers, so that either Variables or Parameters may be used.

The "store" menu option is used to assign values to Parameters and Variables. When this option is executed a prompt appears in the main menu window requesting the letter of a Variable or Parameter. The number on the top of stack will then be assigned to the destination that you specify. For example, to assign the value .707 to the "A" Parameter, first type .707 **Enter**, and then execute **store** from the calculator menu. The following message will appear:

Select a parameter or variable: _

Press A and ".707" will move from the top of stack into the A Parameter. Notice that the display is immediately updated to reflect this change of status. The "store" function can also be executed in a command line. In this case the letter of the target Parameter or Variable should immediately follow the command. For instance, we can use a command line to assign the value of $\sqrt{2}$ to the "R" Variable:

Command: 2.0 sqrt store R

The contents of a Parameter or Variable can be recalled to the top of the stack simply by typing the associated letter. Type R R * and press **Enter**. You will produce the square of "R", i.e. 2.0, on the top of stack.

Let's try an example with a little more meat. Suppose that we want to compute the x and y coordinates of the point where the lines $y=ax+b$ and $y=cx+d$ intersect. These coordinates are given by the formulas:

$$x=(d-b)/(a-c)$$

and

$$y=(da-cb)/(a-c).$$

To make things more specific, we will use the lines $y=2x+3$ and $y=5x+7$. The slopes and intercepts of these lines could be plugged directly into the above formulas, but for demonstration purposes we will first install them in the A, B, C and D Parameters. There is a very easy way to do this without using "store". The "pedit" (parameter edit) menu option allows editing of all the Parameter values at once. Find **pedit** in the menu of Calculator Functions and execute it. A list of the Parameters and their values will be displayed in the middle of the screen, and a flashing cursor will be positioned alongside the first one. [Refer to Screen 1.6]. You can begin by typing in a replacement value for the A Parameter or you can use the arrow keys to skip over Parameters leaving them unchanged. Use "pedit" to assign the values A=2, B=3, C=5 and D=7. When you are done, press **Esc** to get back to the main menu.

Parameter Editing			Parameters
A= (2.0000) B= (3.0000) C= (5.0000) D= (7.0000) E= (.0000) F= (.0000) G= (.0000) H= (.0000) I= (.0000)			A= 2.0000 B= .0000 C= .0000 D= .0000 E= .0000 F= .0000 G= .0000 H= .0000 I= .0000
Press Esc when done			Variables
Acquire Graphics Functions File Proc File I/O Polys Wave Proc Curve Fit Diff Eqs Wave Gen Stats Save/Exit			R= 1.4142 S= .0000 T= .0000 U= .0000 V= .0000 W= .0000 X= .0000 Y= .0000 Z= .0000

SCREEN 1.6

Now to compute the x and y intercepts of $y=2x+3$ and $y=5x+7$ we can use the above formulas in command lines. Remember to enter \ as the first character in each line to indicate that you will be using algebraic syntax. To compute the x-coordinate enter:

Command: \ (D-B)/(A-C)

The resulting x-value, -1.3333, is pushed to the top of the stack. To compute the y-coordinate enter:

Command: \ (D*A-B*C)/(A-C)

The value of Y, 0.3333, is pushed on top of the previously calculated X value.

At this point you may find it helpful to begin using the line editing features of the ASYSTANT. Each time you enter the command mode, the preceding command line can be automatically re-entered by pressing the F3 key. F3 will work this way only after you have entered the command line, so be sure to enter the first character of the line you want, and then hit F3 to bring back the previous line. Also, the arrow keys can be used to move the cursor backward and forward through the line without erasing characters, and text can be inserted or overwritten starting at the cursor position. The "Ins" key will toggle between the insert and overwrite mode. This line editing capability makes it very easy to repeat calculations using different parameter values, or slightly different formulas.

1.5 Defining User Functions

You can make the Desk Calculator even more powerful by defining your own functions. The F1, F2, ..., F10 function keys can each be assigned a user function, a sequence of up to 5 command lines. (If you create a user function for the F3 key, it will still function as a line editing key in the command mode. When you exit the command line, F3 will resume its defined meaning.) To specify these command lines you must first return to the Main Menu (press **PgDn**), and then execute the **Functions** menu option. The stack contents window will then be erased and repainted with a list of the 5 command lines assigned to the F1 key. Initially, of course, they will all be blank. The "PgUp" and "PgDn" keys will let you switch the display over to the command lines for the other Function Keys.

A flashing cursor will be displayed at the beginning of the first command line and you may start entering text at this point. The arrow keys will move the cursor backwards and forwards as well as up and down, without disturbing the text. Also, the "Home" and "End" keys will move the cursor to the beginning of the first and last lines. Text can be inserted or overwritten at the cursor position, and the insert/overwrite mode can be toggled back and forth by the "Ins" key. Text may be erased using the backspace key.

As an example, we will set up the F2 key so that it will compute the point of intersection of two lines, as described in the previous section. First press the **PgUp** key so that the display will be setup for editing the F2 function. Now enter the following text:

```
F2  1> \ (D-B)/(A-C)
      2> store X
      3> X
      4> \ (D*A-B*C)/(A-C)
      5> store Y
```

We have changed the procedure a little so that the solution values are saved in the X and Y Variables. The X in line 3 ensures X will also be left on the top of the stack -- this will save us some keystrokes later.

Press the **Esc** key. If you entered only valid ASYSTANT commands, you will return to the main menu. At this point the command lines have been compiled into the F2 function. However, if an error was encountered in compilation, it will be reported on the display and you will have an opportunity to re-edit the function. Make the changes and try **Esc** again, until the function compiles successfully.

To try out the function, first press **PgUp** to get back to the calculator. Next use **pedi**t to assign any values to the A, B, C and D Parameters. Finally, press **Esc** to leave **pedi**t, and F2 to carry out the operation, storing the results in X and Y, with a copy of X left on the stack. The F2 function can also be executed in the command mode by typing in the characters "F2".

Try computing the point of intersection for another pair of lines; it now only takes two steps. First set up the coefficients using **pedi**t, and then press F2.

1.6 The ASYSTANT's Notepad

You may now wish to make some notes about our recent calculations. Suppose, for example, that our equations have represented the cost/profit relationship for a new invention. The cost function is a straight line $y=ax+b$, where b is the overhead or fixed cost, and a is the production cost of each unit. The revenue function is also a straight line $y=cx+d$, where $d=0$ and c is the price per unit. The x -coordinate of the intersection of these two lines is the number of units that must be sold in order to break even.

In light of this new information, the calculations take on an added significance. In the ASYSTANT's notepad you can record this information as you compute.

To invoke the notepad, press **Ctrl-N** (i.e. hold down "Ctrl" as you would a shift key and press "N"). The display will then be completely erased, and the following message will be written at the bottom:

Default file is: NOTES.DOC	F9 Directory
Type file specification or <cr>:	F10 Exit

The filename "notes.doc" is used initially as the default for the notepad, but you may wish to specify a different name that is more relevant to the present calculations. The file can be made to reside on any drive in the system by including a drive specifier in front of the file name. The "F9" key will produce a directory listing, and the "F10" key will return you to the calculator.

If the specified file already exists it will be read in from disk for editing. If it does not exist then a new file will be created. Right now let's create a new file called **brkevn.doc** (short for break-even documentation). Type in this filename followed by the **Enter** key.

The bottom portion of the display will be repainted with a menu of edit options. This menu contains a number of high level editing features that are explained in Chapter 1 of the Reference. Text can be entered at the cursor by typing it directly in at the keyboard. It can be inserted or overwritten, and the insert/overwrite mode can be toggled by the "Ins" key.

Try typing in the following brief set of notes:

Assumptions: Fixed cost of producing the Invention is \$150,000
Incremental production cost is \$30 per units

The table below outlines the number of units that must be sold to break even, for various selling prices:

Units	Price (\$)
-------	------------

Now press the F9 key to save these notes on disk. Press F10 twice to leave the editor. The first F10 takes you back to the file specification prompt, and the second returns you all the way to the calculator.

Let's calculate the number of units need to break even for selling prices of \$400, \$500, and \$600 using the F2 function we defined earlier.

To find the break-even value for \$400, we must find the x- coordinate of the intersection of the lines $y=30x+150000$ and $y=400x+0$. The F2 function will solve this problem for us once the A, B, C, and D Parameters have been assigned the proper values ($A=30$, $B=150000$, $C=400$ and $D=0$). Enter the specified values using **pedit** (remember to press Esc to exit the "pedit" menu). Then press the F2 key. The stack contents should be:

405.4054 (From REAL SCALAR X)

Repeat this process twice, using **pedit** to enter 500 and then 600 in Parameter C before pressing F2 each time. After you're done, the stack should contain:

263.1579 (From REAL SCALAR X)

319.1489 (From REAL SCALAR X)

405.4054 (From REAL SCALAR X)

Note that as you change X, the previous value you placed on the stack remains, but is pushed down.

You now have a choice between two methods for storing your calculations in the note pad. The first is simpler, but does not demonstrate the full power of the "print" option, which automatically writes data to the notepad, as well as the screen or printer. The second more powerful method demonstrates "print" and also some of the ASYSTANT's array manipulation capabilities. We can create a table of values in an ASYSTANT array, and then automatically "print" the completed table to the notepad. This method, however, requires operations which are introduced in more depth in the next session. Users who are familiar with array operations in computers will have no trouble working through this example and should skip now to section 1.8, METHOD 2. New users may wish to work through the simpler method now in section 1.7 and then return to this example after they have completed Session 2.

1.7 METHOD 1 - Entering numbers into the Notepad

We will enter each stack entry in turn into the notepad by selecting **print** in the menu of calculator functions. [Refer to Screen 1.7]. "Print" will write the top stack entry to a chosen output device. Select **D** for disk output to the notepad file, **Y** to continue, and **N** to append the data to the end of the file, rather than insert it at the cursor location. Then press any key to return to the calculator. Drop the value just entered into the notepad off the top of the stack by selecting **drop** in the menu of calculator functions.

Now repeat this process for the next two values on the stack. If you then press **Ctrl-N** and **Enter** you can see your updated notepad file. The numbers will have been entered in a column under "Units" (you may have to reposition the column heading slightly). You could now manually enter the prices \$400, \$500, and \$600 for each row of your table if you wish.

Please continue with section 1.9. If you wish to try METHOD 2 now, you first need to set up the stack and note file again as you did in section 1.6.

1.8 METHOD 2 - Creating a Table for the Notepad

We will use ASYSTANT's array manipulation capabilities to create a 2 dimensional Variable array to hold a matrix of Unit and Price values and then use the "print" option to write it in the note file. The table will have the three prices in the right hand column, and the corresponding breakeven volumes in the lefthand column. The final result, therefore, will have three rows and two columns.

ASYSTANT gives you many operators to allow you to shape your data exactly the way you need it. To illustrate these features, we will create two 3-element vectors, one of prices, the other of unit volumes. We will then laminate the two to create a single array and automatically print the completed table in the note file.

The stack should currently contain:

```
263.1579 ( From REAL SCALAR X )  
319.1489 ( From REAL SCALAR X )  
405.4054 ( From REAL SCALAR X )
```

We will create the vector of unit volumes by combining the top three entries on the stack into a single array. Press the **Home** key to move to **next** in the Calculator Functions menu and press **Enter**; you will see the Array Operations menu appear. There are actually 4 menus here, which you can page through by repeatedly selecting "next". We will cover these menus in greater detail in the next session. Now select **cat** from the Array operations menu. The top two entries will disappear, replaced by:

```
Unnamed Array REAL DIM[ 2 ]
```


This contains the numbers: 319.1489 , 263.1579. Select **cat** again, and the stack will contain:

```
Unnamed Array REAL DIM[ 3 ]
```

We now have 3 element unit volumes vector.

To create the price vector, we use the "n:ramp" option on the Array Operations menu. Press **3 Enter** to put the number 3 on top of the stack, and then select **n:ramp**. The Stack Contents window now displays:

```
Unnamed Array REAL DIM[ 3 ]
Unnamed Array REAL DIM[ 3 ]
```

You have just created a "ramped" array of three elements. Were you to print this array to the screen (you can do this now if you want - select **next** three times to page back to the Calculator Functions Menu and then choose **print**), you would find it contains the values:

```
1.000 2.000 3.000
```

In general, **n:ramp** creates a new vector. The dimension of the vector is specified by a number on top of the stack, and each element is set to the value of its index. "100 **n:ramp**" therefore would create a new 100 element vector containing the values 1,2,...100.

If we add "3" to every element of the vector, we'll end up with the values 4.00, 5.00, 6.00. Then, if we multiply each entry by 100, we'll have the price vector we want. We can perform this calculation in one command line in the ASYSTANT:

```
COMMAND: 3 + 100 *
```

Do this now, remembering to put a space between each entry. ASYSTANT will automatically add 3 to *each element of the array* on the top of the stack, and then multiply each element by 100. This ability to operate on every element of an array is one of the most powerful features of the ASYSTANT, which we will explore in much greater depth in later sessions.

Store the result in Variable V by typing: **STORE V**. (Remember to press **Enter** after each line). Now type **V Enter** to put a copy of the values just stored back on the stack. The stack display should read:

```
V DP.REAL DIM[ 3 ]
Unnamed Array REAL DIM[ 3 ]
```

We now have two 3 element vectors, the second one holding the breakeven volumes corresponding to the price levels contained in the top vector. To fuse them together, move the cursor one item to the right and select **lam**. The stack now contains:

```
Unnamed Array DP.REAL DIM[ 2 , 3 ]
```

"Lam" has laminated the two arrays, but the columns and rows are transposed from what we want. Printing this matrix would give us:

```

405.4054    319.1489    263.1579
400.0000    500.0000    600.0000

```

To fix it, we can use the "trans" operator in the upper right corner of the menu. "Trans" (for transpose) will cause Row 1 to become column 1, and row 2, column 2. Select **trans** and note that the row and column indices are transposed on the stack:

```
Unnamed Array DP-REAL DIM[ 3 , 2 ]
```

This is our table! Press **Home** and select **next** three times to page to the menu of Calculator Functions again. To enter this into the note file, execute the **print** menu option. A window will appear in the upper left corner. [Refer to Screen 1.7]. It will ask you if you wish to direct output to the disk, printer, or screen. Disk output refers to the note file, so enter **D**. When **D** is selected and you enter **Y** to continue, you will be asked if the output is to be inserted at the current cursor location, or if it is to be appended to the end of the file. Type **N** to insert the table at the end of the file. The disk I/O will take a few seconds to execute, after which the data values will be neatly inserted. You can confirm this by returning to the note file using **Ctrl-N**.

Print			Parameters
Out to Printer, Disk or Screen: (D)			A= 30.0000
Continue (Y or N): (Y)			B= 150000.0000
Insert text at cursor (Y or N): (N)			C= 600.0000
			D= .0000
			E= .0000
			F= .0000
			G= .0000
			H= .0000
			I= .0000
			Variables
			R= 1.4142
			S= .0000
			T= .0000
			U= .0000
			V= 3 DP-REAL
			W= .0000
			X= 263.1579
			Y= 157894.7344
			Z= .0000
Acquire	Graphics	Functions	
File Proc	File I/O	Polys	
Wave Proc	Curve Fit	Diff Eqs	
Wave Gen	Stats	Save/Exit	

SCREEN 1.7

1.9 DOS Operations

Another invisible feature of the ASYSTANT is the ability at any time to invoke certain DOS operations. These let you list and manipulate the files that are currently on disk. Right now we will use the DOS Delete operation to erase the file BRKEVN.DOC, since it was created only for demonstration purposes.

Press **Ctrl-D** to call up the menu of DOS operations. (Hold "Ctrl" down like a shift key while you hit "D"). The screen will then be erased temporarily and a new menu will appear in the upper left hand corner. [Refer to Screen 1.8]. Select the **Delete File** option. Prompts will then appear asking for the file specifications: enter **d:BRKEVN.DOC**, where **d** corresponds to the drive identifier where the file is located. Then enter **C** in the next prompt, to tell ASYSTANT that you wish to continue. The file will then be deleted. The continue/quit option is used in many parts of the ASYSTANT as a kind of fail safe that allows you to terminate operations. Had you decided not to delete BRKEVN.DOC, you could have entered "Q" instead of "C", or simply pressed "Esc" or "Enter" to accept the default "Q".

Operations	Work Space
Delete File Copy File Rename File Directory Return	File(s) to delete: <C:BRKEVN.DOC> Quit/Continue: < C >

SCREEN 1.8

You now have the option to use "up arrow" to specify another file to delete, but in this case hit **Enter** to get back to the "Operations" menu. You can then use the **Directory** option to confirm that the file has actually been erased. Finally, select **Return** to get back to the Desk Calculator.

1.10 Getting Help

An important feature of the ASYSTANT is the on-line help system. This system allows you to get help at any time, just by pressing the "?" key. If you are working on a floppy based system, you will need to insert the Help diskette in the drive you specified for the help file when you configured ASYSTANT (usually drive B -- see section 1.1.7). When you are through perusing the help file, you *should return the previous disk to the drive.*

The help system is set up as a tree that is organized along the same lines as the Reference section of the manual. Where you enter the tree will depend to some extent on where you are when help is requested. For example, while in the calculator menu, hit ?. The screen will temporarily erase, and a description of the functions of the calculator menu will appear. [Refer to Screen 1.9]. To return to the Desk Calculator, hold down the Alt key (as you would a shift key) and hit F10. ASYSTANT will restore the screen and return you to the Desk Calculator.

You can type "?" at virtually any point in the operation of ASYSTANT, and return exactly where you left off.

This menu contains options for stack manipulations, standard elementary functions, and also and several miscellaneous operations like store, prnt, pedit and status. The functions all act on the entry on the top of the stack, and work with all data types including complex numbers. However, radian angular units must be in effect for functions to work properly on complex arguments. When a function is applied to an array, the result is a new unnamed array of the same shape as the original. The values of the resultant array are obtained by applying the function to the original array on an element by element basis.

next

Advances to the next menu: Array Operations

store

Type space bar to continue...

Current: Desk Calculator

AltF10 - Exit

F1 Back to: Asystant

F2 Calculator Overview

F7 Conversions Menu

F3 Entering Numbers

F8 Using the Command Mode

F4 Basic Arithmetic

F5 Desk Calculator Menu

F6 Array Operations Menu

SCREEN 1.9

1.11 Errors

Unless you are a very accurate typist, you probably have already encountered some errors. They can be generated in many different ways. For example, entering an invalid command will generate an error, and so will dividing by zero. Whenever an error crops up, a special window is displayed in the center of the screen with a message explaining the error. You can then recover from the error by pressing any key.

There are two things to keep in mind regarding errors. The first is that the messages are read from the disk file `ASYSTANT.MSG`, found on the System, the Analysis, and for `ASYSTANT+`, the Acquire 1 and 2 diskettes. If this file is not available, no error message will be displayed, although the error number will still be reported.

Secondly, be aware that sometimes the stack contents will be corrupted by errors. This is especially true of more complex operations that use the stack to retain intermediate as well as final results. This is normally invisible to you because the intermediate results are erased during normal execution. If an error occurs before an operation has a chance to erase these intermediate results, you will find unexpected data on the stack. You can be sure that whatever is left on the stack *is not the final answer, so don't be fooled*. You should also expect the original input parameters on top of the stack to be gone. Thus you should always check the stack after an error before attempting to continue with a calculation, and if you are performing an elaborate calculation, you should be using Parameters and Variables to store intermediate results.

Right now, let's intentionally generate a math error. Enter the following command:

```
COMMAND: 1 0 /
```

After you press **Enter** to accept this command, `ASYSTANT` should display an error message. However, you may find that nothing will happen, and `ASYSTANT` will seem to freeze up. There are two possible explanations for this behavior:

1. You have failed to set a DIP switch on the motherboard of your computer indicating that the Numeric Coprocessor is installed. In this case the coprocessor will function, but it cannot return errors to `ASYST`. Check with the dealer who installed your coprocessor, or refer to the appropriate hardware reference manual for your computer.

2. If you are running ASYSTANT on an IBM Compatible, the BIOS ROM in your machine handles coprocessor errors incorrectly. After you check the dip switches, and if you are still having problems, call Macmillan Software's Technical Hotline to report this problem. Manufacturers frequently update their BIOS ROM's without announcing the updates: if this is the cause of your problem, our service personnel may be able to help you to order a free or low-cost replacement ROM from the manufacturer that fixes the problem.

Please keep in mind that Macmillan Software cannot guarantee the performance of the ASYSTANT on all so-called compatibles. If you are considering buying a new compatible you may wish to check whether it is supported by Macmillan Software. While ASYSTANT operates on many machines, performance is guaranteed only on supported systems.

Whatever the cause of your problem, hitting "Ctrl-Break" will usually unlock the keyboard and return you to the Desk Calculator.

1.12 Saving Things and Saying Goodbye

We've now come to the end of our first session. To leave the ASYSTANT, go back to the Main Menu (using **PgDn**) and select the **Save/Exit** menu option. The screen will clear, and the following prompt will be displayed:

Save/Exit/Return: <E>

Save parameters, variables and functions
Exit to DOS
Return to the desk calculator

If the "Save" option is selected you will be prompted for a filename in which to save the current contents of the parameters, the Variables and any user functions you defined. This information can be recalled the next time that the ASYSTANT is used, so that you can resume your calculations exactly where you left off. There is no need to save anything this time, so just press Enter and you will go directly back to DOS.

SESSION 2: **ARRAYS AND GRAPHICS**

In this session we will begin to use array operations and examine some of the ASYSTANT's plotting capabilities. The following features will be used:

Desk Calculator

to create and do arithmetic with arrays;

Graphics

to generate simple plots,

to plot in different graphics windows,

to label plots;

Mini Calculator

to perform calculations away from the Desk Calculator menu.

One of the most powerful features of the ASYSTANT is its ability to manipulate arrays as if they were single numbers. Arrays can be put on the number stack as single entries, and added, subtracted, multiplied, etc. by the same operations that are used for numbers. Expressions can mix numbers (i.e. scalars), vectors, and two dimensional arrays. Furthermore, there is a powerful array editor that lets you easily create and modify arrays. In latter sessions we will also see how arrays can be edited graphically (see Session 4).

2.1 Array Arithmetic

To get things started, boot up the ASYSTANT using the procedure described in the previous session. This time, remember, the configuration step is unnecessary.

Press the **PgUp** key to get into the calculator, and execute the **next** menu option to switch to the menu of Array Operations. [Refer to Screen 2.1]. There are four menus of calculator functions; until now we have only needed the first. Each of these menus has a "next" option that will call up the next menu. Try pressing the **Enter** key several times. The four menus will be displayed in succession. You might want to take a minute to scan over the options in all four menus, just to see what is available. The function in each menu item is described in the Reference section of the manual (see Chapter 2), and can also be found in the Help system.

Stack Contents				Parameters	
(empty stack)				A=	.0000
				B=	.0000
				C=	.0000
				D=	.0000
				E=	.0000
				F=	.0000
				G=	.0000
				H=	.0000
				I=	.0000
Array Operations				Variables	
next	xsect	sub	trans	R=	.0000
aedit	diag	n:ramp	nm:ramp	S=	.0000
reverse	n:rot	cat	lam	T=	.0000
[]sum	[]prod	[]max	[]min	U=	.0000
^sum	^max	^min	reshape	V=	.0000
sort	index	lookup	n:search	W=	.0000
Main Menu Options				X=	.0000
Acquire	Graphics	Functions		Y=	.0000
File Proc	File I/O	Polys		Z=	.0000
Wave Proc	Curve Fit	Diff Eqs			
Wave Gen	Stats	Save/Exit			

SCREEN 2.1

Now let's try generating an array. Switch back to the Array Operations Menu on the Desk Calculator and enter the number 100 on the top of stack. Execute the **n:ramp** menu option (row 2, column 3). The stack will now contain a single entry that is an "Unnamed Array REAL" with dimension "DIM[100]". Store the array in Variable R by typing:

Command: STORE R

The command "n:ramp" fills the array it creates with the integers corresponding to the index of each element, i.e., 1.0, 2.0, 3.0, 4.0, ..., 100.0. You could check this by going back to the menu of calculator functions and using

"print", but it is easier to view the elements using the system's array editor. To invoke the array editor execute the **aedit** menu option (row 2, column 1 of the Array Operations Menu). ASYSTANT will prompt you to select a Variable. "R" will be the default. Hit **Enter** twice, first to accept "R" and then to accept "Continue". The screen will then be repainted with the text:

```
1.0000  2.0000  3.0000  4.0000  5.0000  6.0000
```

```
R  REAL DIM[ 100 ]
```

```
Col:      1
```

```
Enter Value <CR>:
```

The numbers displayed along the top line are the values of the first six array elements. Notice that the first element is highlighted. The value of this element can be changed by typing in a new number in response to the "Enter Value:" prompt. Try typing the number 1.01 and press **Enter**. Notice the highlight automatically moved over to the next element to the right. The arrow keys can be used to move the highlighted element through the array without changing the existing values. Hold down the **right arrow**. Once the highlight hits the rightmost column of the display, the array will begin to scroll, and it will continue scrolling until the right boundary of the array is reached. Now try hitting the **left arrow** at least once, and type in a new value that is .01 different from the current value (eg., 99.01 if the current value is 99). Notice that when you hit **Enter** the highlight moved to the left, not the right as before. This is because you hit the left arrow just before entering the value. This feature makes entering a series of numbers easy. You can start with either the first or the last element. Just be sure to hit the left arrow key before entering the first number in the series, if you want the cursor to continue moving left.

The **Home** key will position the highlight on the first column of the first row; the **End** key puts the highlight on the first column of the last row. Since there is only one row on the screen, Home and End are the same in this case.

Press **Enter** without typing in a number to return to the calculator.

ASYSTANT lets you create new arrays either with "aedit" or "n:ramp" , and "nm:ramp" (for two dimensional arrays). To create an array with "aedit", respond to the entry prompt with the name of a Variable that does *not* contain an array. "aedit" then assumes you wish to create a new one, and gives you a series of prompts to specify dimensions and type. The values in the new array will be initialized to zero. The two ramp functions initialize the array with successive integers, which can be very useful. For example, suppose that you want to plot the function $y=2x^2+3$ over the interval [1,2]. You should first produce an array of x-values that are equally spaced, and span the interval [1,2]. "n:ramp" makes this very easy, which is why we used it instead of "aedit".

Starting off with our original array of 1,2,...,100, we would divide each element by 100, and then add in a 1. This gives us a new array that contains the values 1.01, 1.02, ..., 2.00.

Note: If, while in the array editor, you changed any element by more than 0.01, you might want to regenerate the array by typing:

Command: 100 N:RAMP STORE R

The following command line will carry out the math and store the new array in X:

Command: R 100. / 1. + STORE X

Once again you can verify that this gives the correct result with **aedit**. Note that ASYSTANT has carried out the required mathematics on every element of the original array! This is one of the most powerful features of the Desk Calculator, and will save you significant time compared to a conventional scientific calculator.

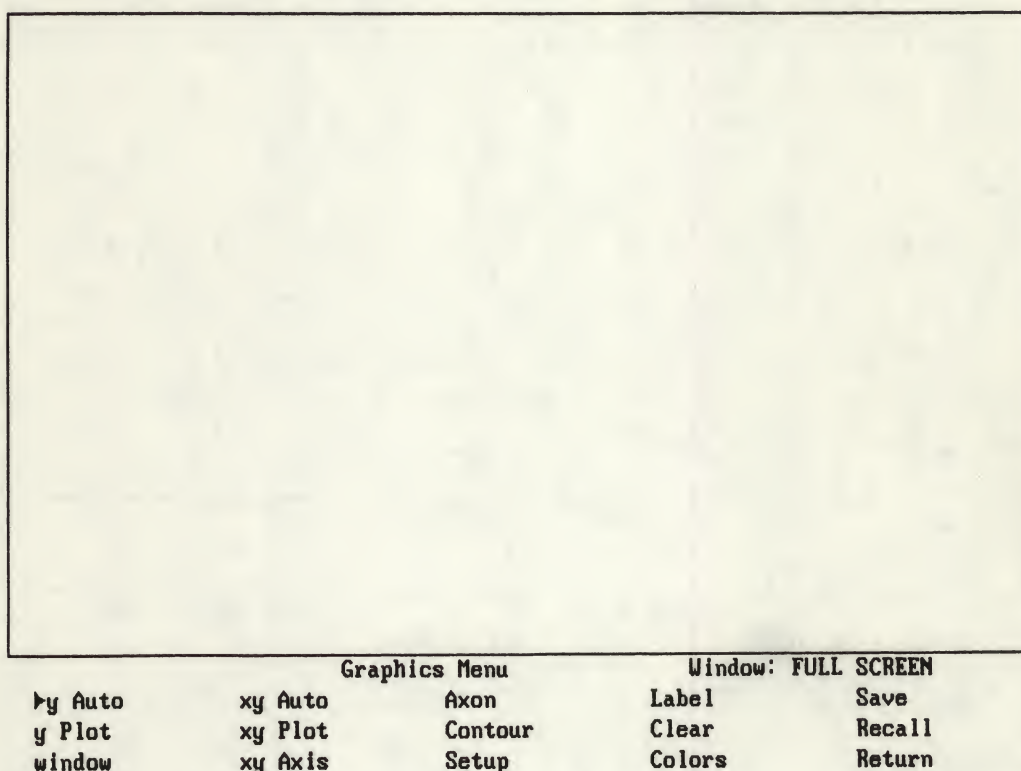
Now generate the array of y-values as a function of X array values. This can be done with a single simple algebraic expression:

Command: \ 2*X^2+3

This applies the quadratic polynomial to each element of X and generates the result on the stack. If we now store this result in the Y Variable, (**STORE Y**), we will be ready to make some plots.

2.2 Generating Simple Plots

To get into the Graphics menu, press **PgDn** to return to the Main Menu and execute the **Graphics** option. Remember, if you are running a floppy system, the yellow Analysis Diskette must be in the System Overlay drive (drive A in this tutorial) in order to execute the Main Menu options. The Desk Calculator will disappear, replaced by a large blank window occupying most of the screen with the Graphics Menu at the bottom. [Refer to Screen 2.2.] The blank area is used for displaying graphics. You can direct plots to any of the following nine windows that sub-divide this area: full screen, top half, bottom half, right half, left half or one of the four quadrants. The "window" menu option allows you to change the plotting window, and the currently active window is always displayed on the right side of the menu title line.

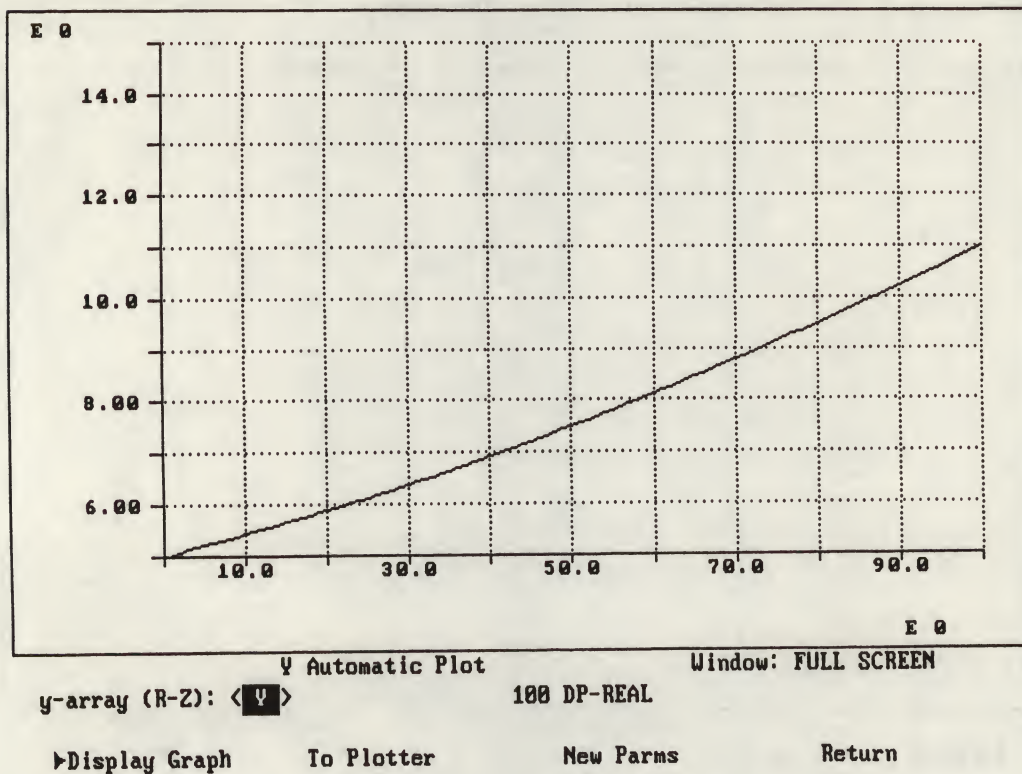


SCREEN 2.2

Note that instead of a highlight, the current menu option is indicated by a small pointer. The arrow keys now move the pointer, just as before they moved the highlight, and "Enter" will accept the current choice. A pointer is used because ASYSTANT has switched your computer to graphics mode, and most graphics cards will not support highlighted text in graphics mode.

Execute the **y Auto** option. This will produce a plot using an automatic scaling procedure to compute the x and y axes. A prompt will appear in the menu window asking for the name of the Variable to be plotted. In this instance we want to plot Y, so enter **Y**, and then **Esc** to continue.

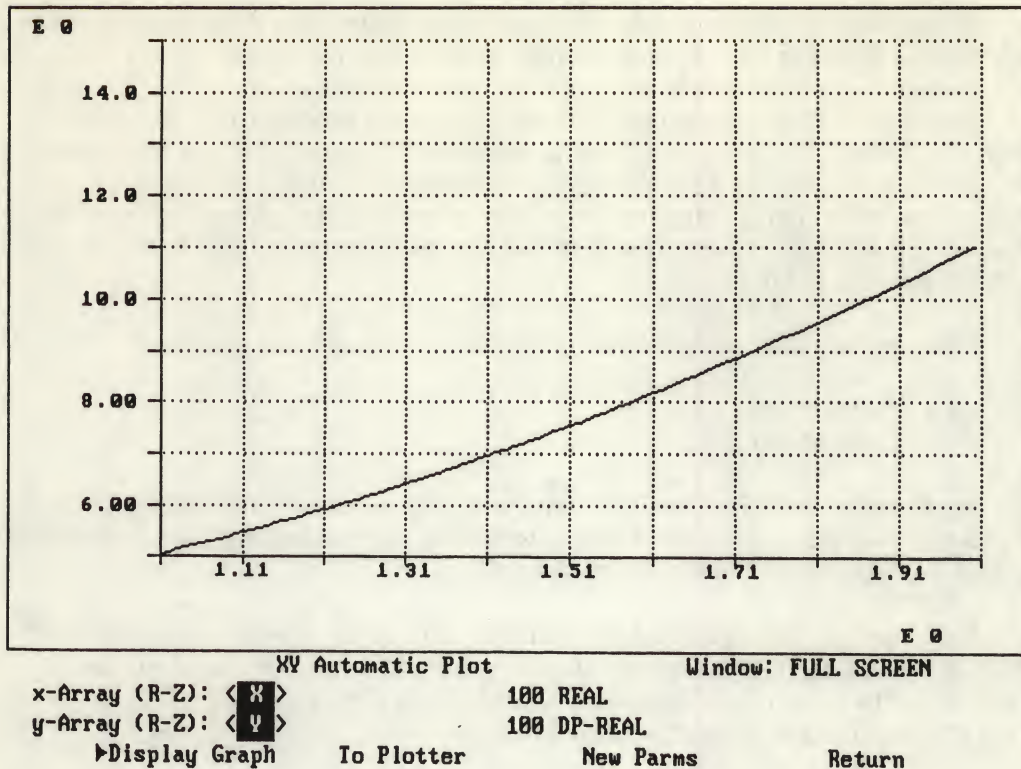
A one line menu will emerge at the very bottom of the screen. Select **Display Graph** to draw the plot [refer to Screen 2.3]. Other options in this menu allow for output to the plotter and for selection of a different plot array. Execute the **Return** option to go back to the main Graphics Menu. (A fast way to get to the "Return" option is to hit the **End** key on the keypad.)



SCREEN 2.3

Notice that the plot contains correct y-values, but x-values ranging from 1 to 100 rather than from 1 to 2. This is because the array of x-coordinates was not specified. The **y Auto** option only accepts a y value, and always takes the x-coordinates to be the indices of the specified array. (This is also true of the **y Plot** option used to superimpose plots -- see Reference Chapter 4).

Try recreating the plot using the **xy Auto** option. The procedure is exactly the same, except prompts request Variables for both x and y values. Enter **X** for this array and leave "Y" as the y array. You will notice that the result is now correctly scaled. [Refer to Screen 2.4]



SCREEN 2.4

You can modify many plotting features by using the "Setup" option on the Graphics Menu. It will allow you to turn off grids, axes, or labels, to use log scales, or to reposition the axes. It also lets you input your own ranges for the x and y axes to make plots without using the automatic scaling process (see "y plot" and "xy plot", Chapter 4). **Return** from "xy Auto" and execute **Setup**. A long list of prompts is displayed in the menu window. The first allows you to toggle between editing x-axis and y-axis parameters. You could try setting up the y-axis so that it uses a log rather than linear scale, and then replot the X and Y arrays. Log scales are often used for improving dynamic range.

If you have an IBM Enhanced Graphics card you also might like playing around with the **Colors** option. It allows you to assign colors to four different components of the plot. A miniature plot is continually updated in the menu window to illustrate the effects of color changes. Try reproducing the plot with a few different color choices. Color is not supported with the IBM Color/Graphics Adapter. With that system, horizontal resolution in color is only 320 pixels: too coarse for scientific graphics. However, the **Colors** option is still useful when directing output to a color plotter.

Please note: Colors and setup parameters are associated with display windows. When you select a new window the display parameters are switched to those that were most currently specified (or defaulted) for the new window.

To generate more interesting plots we must create new data with the Desk Calculator. Execute the **Return** option in the Graphics Menu.

This time let's try something that's a little more challenging: plots of lissajous figures. A lissajous figure is the path followed by a two dimensional harmonic oscillator. To be more specific, imagine an object sitting on a frictionless table that is constrained in both directions by springs. If the object is displaced from its equilibrium position and then released, it will follow the path of a lissajous figure.

The parametric equations used to prescribe lissajous figures are:

$$x = a \cdot \cos(b \cdot t + c)$$

$$y = d \cdot \cos(e \cdot t + f)$$

In our example, the "frequencies" b and e would be determined by the spring constants, and the a , c , d and f terms would be determined by initial conditions (x and y positions and velocities at time $t=0$).

Now suppose that we want to track the path of the object for a period of 10 seconds, sampling its position at intervals of .05 seconds. We first set the **T** Variable to be an array of length 200, containing the values .05, .1, .15, ..., 10. This is accomplished with the single command line:

```
Command: 200 n:ramp 20. / store T
```

The next step is to define the **F1** key function so that it will compute the x and y values of the figure and save them in the **X** and **Y** Variables. Execute the **Functions** option in the Main Menu, and enter the following:

```
F1  1> \ A*cos(B*T+C)
    2> store X
    3>
    4> \ D*cos(E*T+F)
    5> store Y
```

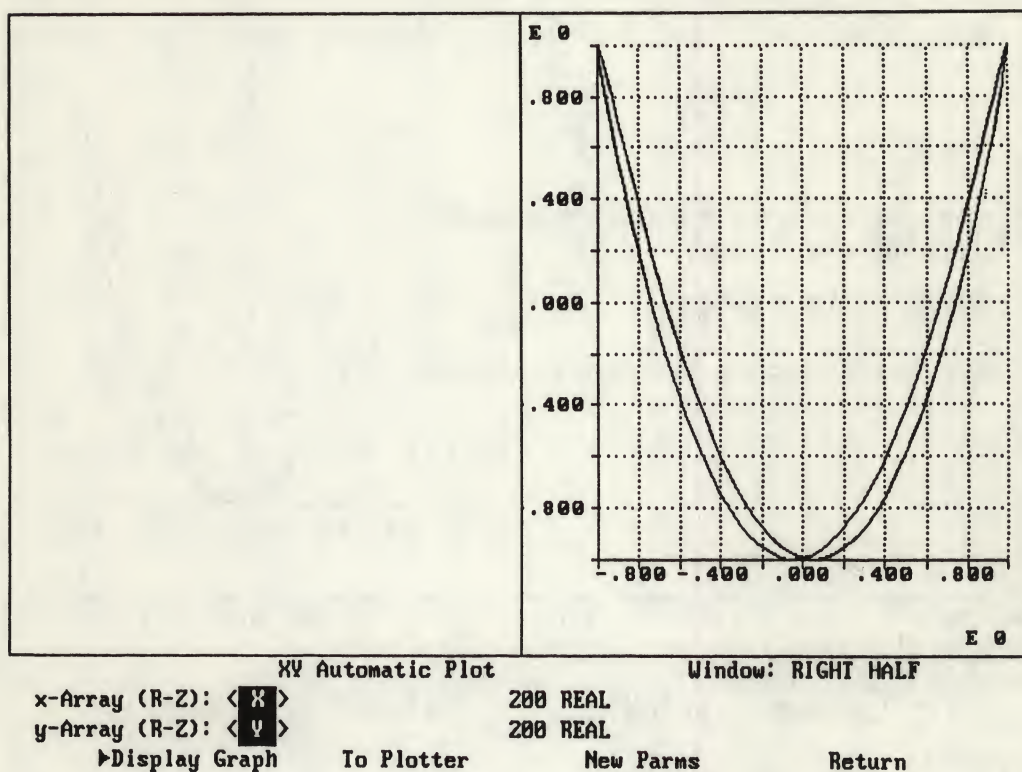
On completion, press the **Esc** key to return to the Main Menu. Then return to the Desk Calculator by typing **PgUp**.

Now use **pedit** on the "Calculator Functions" menu to assign reasonable values to the **A**, **B**, **C**, **D**, **E** and **F** Parameters. For example, try $A=1$, $B=1$, $C=.2$, $D=1$, $E=2$, $F=.5$.

The next step would be to go ahead and press the **F1** key, except that we would like to use radians rather than degrees for the angular units in the trigonometric calculations. The **ASYSTANT** uses degrees by default, but this can be changed via the **status** menu option in the basic calculator menu. (It's just to the right of "pedit".) Type **R** next to the fifth prompt down, for the selection of angular units. Then press **Esc** to return to the main menu.

Now press the **F1** key, and note that the **X** and **Y** Variables contain arrays of length 200. These correspond to the **x** and **y** coordinates of a lissajous figure.

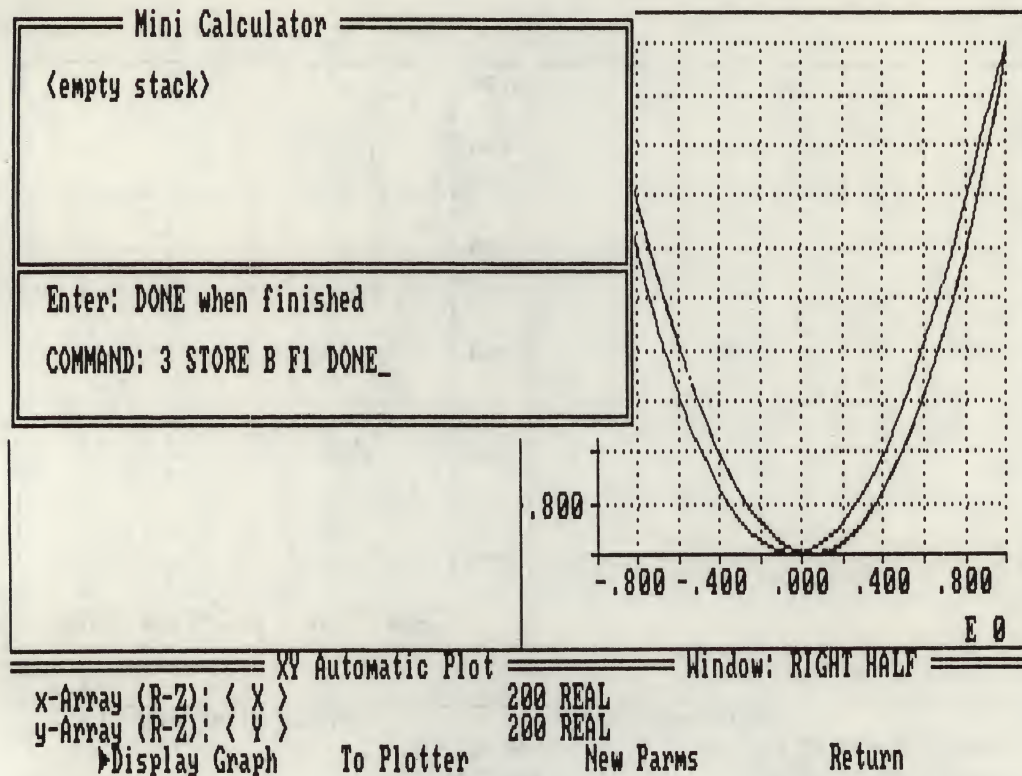
This time, let's plot the figure in the right half of the screen. Hit **PgDn** and execute the **Graphics** option. In the Graphics Menu, select **Window**, and **Right Half**. Notice that the window indicator above the menu has changed from "Full screen" to "Right Half". Now select the **xy Auto** option. Enter **X** for the **x**-prompt and **Y** for the **y**-prompt, hit **Esc**, and execute **Display Graph**. The Lissajous figure will appear in the right half of the graphics window [refer to Screen 2.5].



SCREEN 2.5

2.3 The Mini Desk Calculator

After you have plotted the original figure you might like to try a new set of parameters. There is an easy way to do this without having to exit from the Graphics Menu. Simply press the **space** bar while in graphics, and you will invoke the ASYSTANT's Mini Desk Calculator. The upper left hand corner of the display will be repainted with a list of the current stack contents and a special command line window will appear [refer to Screen 2.6].



SCREEN 2.6

You may now enter commands just as if you were in the main calculator. Keep in mind that some calculator functions cannot be executed from the Mini Calculator (eg., **print**, **status**, **aedit** -- these need additional windows to work).

Let's use the Mini Calculator to change the lissajous parameters (eg., $B=3$) and reexecute F1. We can then plot the modified figure in the left hand window to compare the two plots easily. First, enter the command:

Command: 3 STORE B F1 DONE

The "DONE" command is used exclusively in the Mini-calculator to indicate that you are finished calculating and wish to resume your former activities.

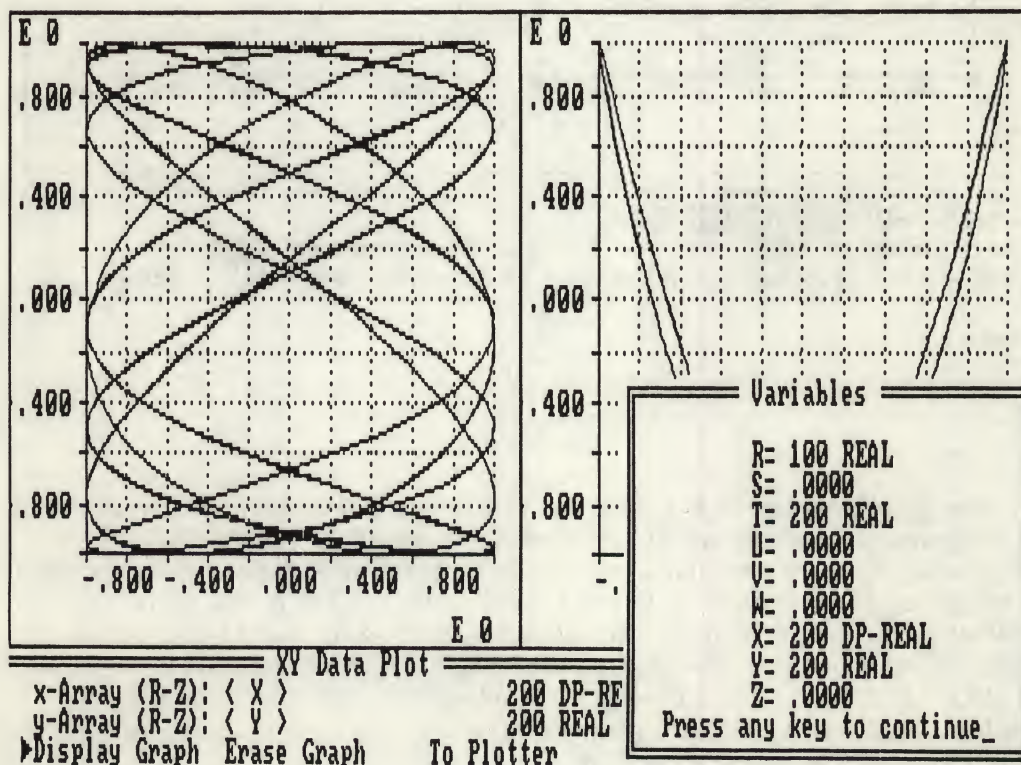
When "done" is executed the screen is restored to what it was before the Mini Calculator was invoked. Note that we used the store command because "pedit" is not available from the command line.

The X and Y Variables now contain the coordinates of the new figure. To plot, select **Return** to get back to the main Graphics Menu. Select **Window** and **Left Half**. Now the new figure can be replotted next to the old one using **xy Auto** as before. Try using the Mini Calculator again, this time with B=5. You can superimpose the new plot on the old one by using the "xy Plot" option. After returning from the Mini Calculator, select **Return** to get to the main graphics menu, and select **xy Plot**.

Please note: So far we have come across four special key operations:

? -- Help
 Ctrl-N -- The Note Pad
 Ctrl-D -- Dos Operations
 Space -- The Mini Calculator

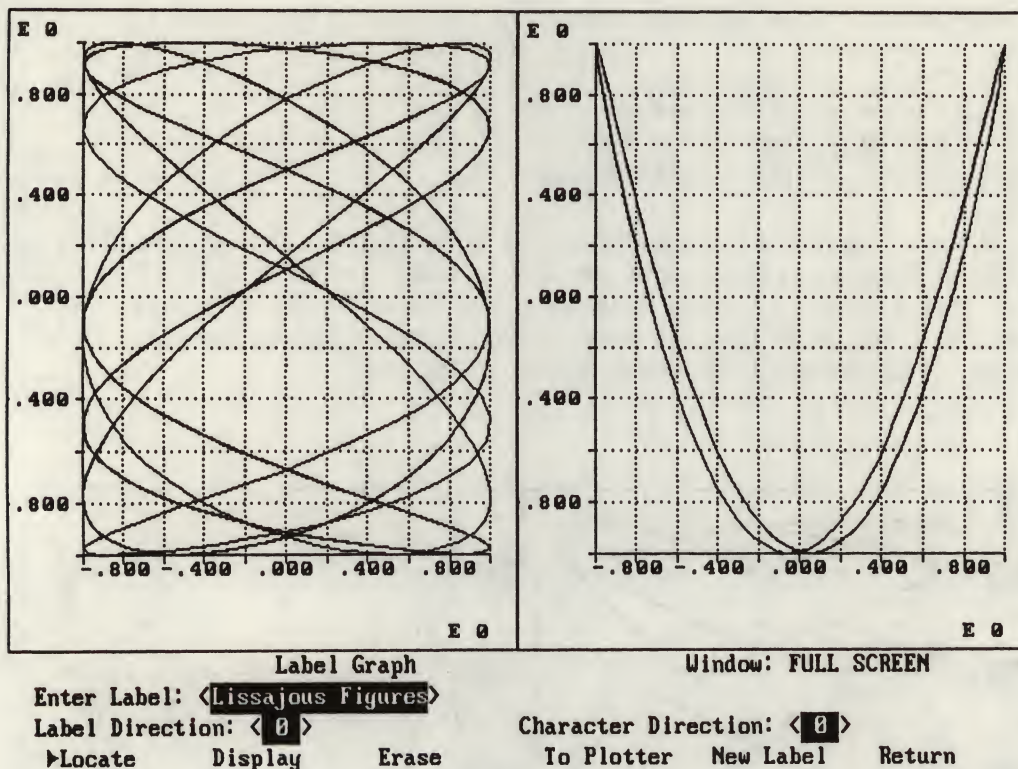
Another operation to be added to this list is **Ctrl-V**. Try it! It calls up a listing of the current contents of the R-Z Variables [refer to screen 2.7]. This is often useful, especially when in the Graphics Menu, to see what arrays are available. All special keys are active in other menus besides Graphics, in fact they are available almost anywhere in the ASYSTANT.



SCREEN 2.7

2.4 Labeling Plots

Let's suppose that you want to label the existing plots. For example, suppose you want to call the overall plot "Lissajous Figures", and label the axes "X-values" and "Y-values". You can do this by using the "label" option in the main Graphics Menu. Select **Return**. Select **Window**, and **Full** so that we can label anywhere on the screen. Now select **Label**. A box will appear on the screen, prompting you to enter a label, and to specify the label and character directions. The three labels listed above must be entered one at a time, so begin by typing in "Lissajous Figures" [refer to Screen 2.8].



SCREEN 2.8

The label direction is the direction in degrees that a label will be displayed on the screen. Thus, if the label direction is 0 the label will be displayed horizontally from left to right, and a label direction of 180 degrees will cause it to be displayed from right to left. If the direction is 270 it will be typed vertically from top to bottom. To display a label on the screen, only label directions of 45 degree increments starting at 0 degrees are allowed, i.e., 0, 45, 90, 135, 180, 225, or 270. Entering any other value will cause the label to display left-to-right, i.e., at 0 degrees. If you are writing to the plotter, label direction can be assigned arbitrarily.

The character direction determines the orientation of each character. A character direction of 0 corresponds to the standard orientation, and at 180 the characters would be upside down. If you are writing to the display, the only allowable character directions are 90 degree increments: 0, 90, 180 or 270. If you are writing to the plotter, the character direction can be assigned arbitrarily.

For this first label leave both the label and character directions at the default value of 0, and press **Esc** to continue.

The Label menu will appear at the bottom of the screen. To begin, execute **Locate** . This option will call up a cursor in the current graphics window that can be moved using the arrow keys. Use the cursor to locate where you want the label to be displayed. You can control the size of the steps taken by the arrow keys by pressing the **PgUp** key, and then entering in a number. The larger the number the bigger the steps.

When you have positioned the cursor in a reasonable location, press the **Del** key. This will return you to the **Label** menu. Now execute the **Display** option to drop the label on the plot starting at the cursor position. If the result is unsatisfactory, erase the label using **Erase** , and use **Locate** again to reposition the cursor.

Now execute the **New Label** option. This will allow you to enter a new label with new directions. Try the "Y-values" label using 90 for both the label and character directions.

Please note: If you want to label a graph on the plotter, it is usually a good idea to generate the plot on the display first. That way you can make sure of the location before you output it to the plotter.

This brings us to the end of another session. We can use a short cut to get back to DOS (as long as we don't need to save the system configuration). Simply press **Ctrl-E** .

SESSION 3: **A STUDY OF PENDULUM MOTION**

In this session we will introduce the "Diff Eqs" and "Curve Fit" options in the main menu. The session will present a study of the motion of a swinging pendulum and use the following features:

Diff Eqs

to solve a system of differential equations;

Desk Calculator

to generate an estimate of the error term;

Curve Fit

to fit a non-linear equation to the error term.

3.1 Numerical Solutions to Differential Equations

Suppose that a mass m attached to the bottom of a pendulum is pulled to one side through an angle θ_0 , and released. The pendulum will oscillate back and forth, following a trajectory described by some function $\theta(t)$. The differential equation that governs this motion can be found from the law of energy conservation. It says that:

$$\text{Kinetic Energy} + \text{Potential Energy} = 0.$$

In the present case, this equation becomes

$$\frac{1}{2} mv^2 = mgl [\cos(\theta(t)) - \cos(\theta_0)],$$

where g = acceleration of gravity (9.8 m/sec²)
 l = length of pendulum (1 m)
 v = velocity of mass

After v is converted to angular velocity ($v=l*d\theta/dt$), the conservation equation will simplify to:

$$\text{Eq. 3.1} \quad d\theta/dt = \sqrt{2g/l (\cos(\theta(t)) - \cos(\theta_0))}$$

This equation cannot be solved in terms of elementary functions. Its exact solution turns out to be an elliptic integral of the first kind. But the ASYSTANT can quite easily generate a numerical solution using the "Diff Eqs" option in the main menu.

To see how this is done, boot up the ASYSTANT and press **PgUp** to enter the Calculator Functions menu. Next, select **status** and enter **R** for radians at the prompt for angular units. ASYSTANT will now work in radians rather than degrees until the status is once again changed (or until you reload ASYSTANT). Now press **Esc**, **PgDn**, and execute the **Diff Eqs** option. The screen will clear and a prompt will ask you for the dimension of the system of differential equations that you wish to solve. ASYSTANT will handle systems of up to five equations, but right now we are only interested in solving one, so respond with **1** and select **Continue**.

Differential Equations

Dimension of system (1 to 5): $\langle 1 \rangle$
Continue/Quit: * C *

Equations

$$dY/dX = \text{SQRT}(19.6 * (\cos(Y) - \cos(\pi/4)))$$

Initial Conditions

X = $\langle .0000 \rangle$
Y = $\langle .0000 \rangle$

Extrapolation Parameters

Final X = $\langle .5000 \rangle$
Step size = $\langle .0100 \rangle$

Solve/Solve&Plot/New Eqs/Quit: * Q *

Variables X,Y will be overwritten

Press Esc to continue

SCREEN 3.1

Immediately after you enter this response, a window will open and the prompt $dY/dX =$ will appear [refer to Screen 3.1]. You must now enter the differential equation to be solved. It must be entered as an algebraic expression, using the same syntax as would be used following a " $\backslash$ " in a command line (but don't include the " $\backslash$ "). X must be the independent Variable, and Y the dependent Variable. The solution curve will actually be built up as arrays of x and y coordinates. The x coordinates will be retained in Variable X, and the y coordinates in Variable Y.

Referring to Equation 3.1, we will be using Y in place of theta, X in place of t, and a constant (say $\pi/4$ radians = 45 degrees) for theta0. Thus, the equation should be entered in as:

$$dY/dX = \text{sqrt}(19.6 * (\cos(Y) - \cos(\pi/4)))$$

When you finish, press Esc and the equation will automatically compile for later use. If you made a typing error, you may get an error message at this point. In this case, reenter the dimension, if necessary, edit the equation to correct your error, and hit Esc again.

If compilation is successful, new prompts will ask for initial values of X and Y. The X Variable represents time and we can arbitrarily assign it the initial value of 0. Y represents the angle of inclination of the pendulum, and its initial value will also be set equal to 0. Since 0 is the default value for both X and Y, you can simply press Esc to continue.

The final step is to specify the extrapolation Parameters. They determine the interval over which the solution is to be carried out and the refinement of the solution curve. We will try solving for the pendulum motion over a 0.5 second period (Final X), using increments of 0.01 seconds (Step Size). Enter these values now.

Please note: The solution curve is built up progressively, starting with the initial values and proceeding step by step to propagate the solution curve. The method used to pass from point to point is the fourth order Runge-Kutta algorithm. It is very accurate, but not exact. Consequently, errors tend to accumulate and compound in the later stages of the solution. In order to keep this effect to a minimum, the step size should be chosen to be relatively small.

The last prompt in this list allows for four options. You can quit and return to the main menu, define a new equation, generate the solution curve (in the X and Y Variables), or generate the solution and plot it. Enter P to elect the last option.

Once the plot has been displayed, you may save it on disk, output it to a plotter, or return to the Main Menu. If you have enough disk space available, try saving it. You can recall the plot later by using the "recall" option in the Graphics menu.

Please note: Keep in mind two factors when saving plots on disk. First, the only information saved is the pixel values of the image. The data arrays and display setup are not kept. In consequence, plots that are recalled from disk cannot be output to the plotter. Secondly, images stored on disk can be recalled only to windows of exactly the same size as the original. In the case of the "Diff Eqs" plots, the image can be recalled only to the Full Screen window.

3.2 The Approximate Solution

Looking at the solution plot, it is evident that the motion of the pendulum is very nearly sinusoidal. In fact, the pendulum is often used in science classes as an example of simple harmonic motion. But this approximation is valid only when the oscillations are small. To see why, we should differentiate Equation 3.1 with respect to time. This gives:

$$\begin{aligned} d^2\theta/dt^2 &= 1/\{2*\sqrt{2g/l} (\cos(\theta)-\cos(\theta_0))\} \\ &\quad * -2g/l \sin(\theta) d\theta/dt. \end{aligned}$$

By replacing $d\theta/dt$ with the right hand side of Equation 3.1, we then have that:

$$d^2\theta/dt^2 = -g/l \sin(\theta).$$

Since $\sin(\theta)$ is about equal to θ when θ is small, this equation approximates that of the simple harmonic oscillator:

$$d^2\theta/dt^2 = -g/l \theta.$$

This equation can be solved explicitly. It gives the following expression for the motion of the pendulum:

$$\theta(t) = \theta_0 \sin(\sqrt{g/l} t).$$

Let's generate this curve and compare it with the numerically computed solution contained in the Y Variable. We simply need to apply the above function to the X array.

Return to the Diff Eqs Editing menu and from there **Quit** to the Desk Calculator. Enter at the command line:

```
Command: \ (PI/4)*SIN(SQRT(9.8)*X)
```

This will produce the approximate solution as an array on the top of stack. Next, subtract off the numerical solution and save the difference in "R" (remember to insert spaces between each entry on the command line):

```
Command: Y - store R
```

This difference should be thought of as the error in making the approximation. You can take a look at it from the Graphics Menu by executing **y Auto**. Select **Display Graph** for the Variable R and **Return** when you are done examining the error plot.

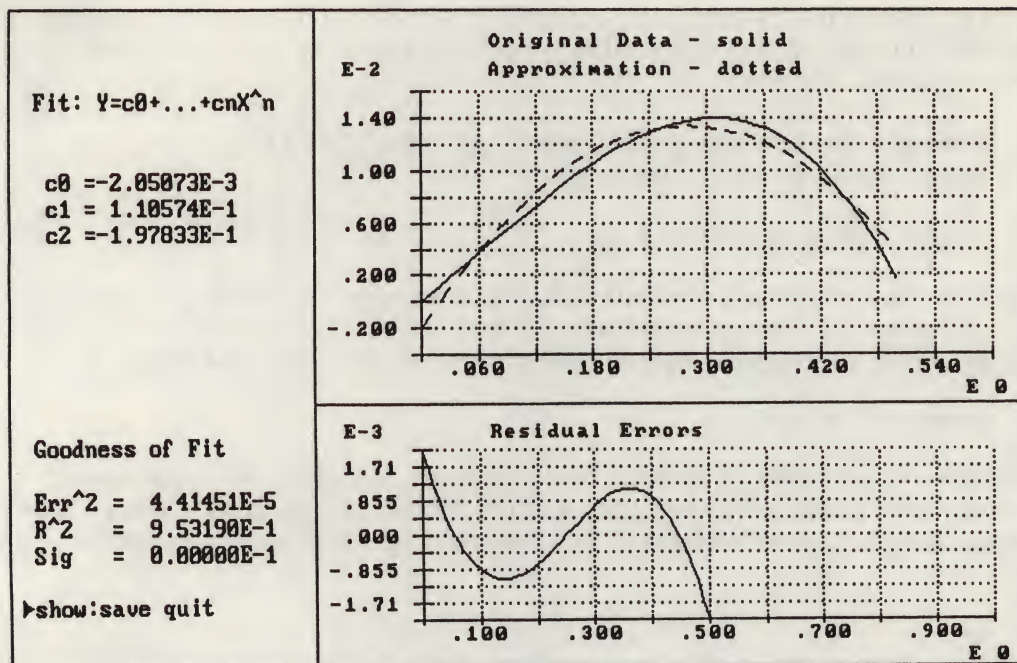
3.3 Fitting a Curve to the Error

It is often useful to model the error term by fitting it to some specific functional form. In this example we will use two different functions to illustrate "Curve Fit". One fit will be to a second degree polynomial and the other to a specific user defined function.

Return to the Main Menu and execute **Curve Fit**. You will be asked to select two Variables, one for the x-values (independent) and the other for the y-values (dependent). Enter Variable X for the x's, and R for the y's. Don't be confused by the default prompts, they will show exactly the opposite of what you want, so you will have to enter the Variables explicitly. Press **Esc** and a selection menu of fit options will be displayed:

```
Linear
Polynomial
Logarithmic
Exponential
Multilinear
User Function
Return
```


Select the **Polynomial** option. Hit **Enter** to accept the default 2 for the degree of the polynomial. This is all that needs to be specified! In a few seconds the display will be repainted with three windows [refer to Screen 3.2]. The upper right portion of the screen will show a plot of the original data with the fitted curve superimposed on it. The lower right will contain a plot of the residual error, and the left side will display a text window listing the polynomial coefficients and several measures of the goodness of fit.



SCREEN 3.2

The "Sig" Parameter reports the significance of the fit. It takes into account the number of Parameters used in the fitting function. Small residuals are less significant if many Parameters are used. The closer "Sig" is to 0, the better the fit. Here, the "Sig" value reported is 0, indicating an extremely high level of significance. See Chapter 9 for more information on all three goodness of fit Parameters.

Before continuing, let's save the results we just calculated. Select **show:save**. After a few seconds, the screen will again be repainted. [Refer to Screen 3.3]. You'll see a prompt list in the upper left window, and other windows will display results: the coefficients and standard deviations, the fit parameters and the correlation matrix. You can save these results in calculator Variables, along with the fitted results and the residuals. Enter **S** by the **Save fitted values** prompt to save the fit in Variable **S**. Then **Esc** to return to the Variable selection prompts. Notice that Variable **S** has been updated.

Save Terms		Coefficients	
Save Fitted values in (R-Z): (S)		Coefficient	Standard Deviation
Save Residuals in (R-Z): (N)		c1 = -2.05073E-3	3.87565E-4
Save Correlation in (R-Z): (N)		c2 = 1.10574E-1	3.58476E-3
Save St. Dev.'s in (R-Z): (N)		c3 = -1.97833E-1	6.93345E-3
Save Poly Coeffs in (R-Z): (N)			
Press Esc to continue ...			
Goodness of Fit			
Errr^2 = 4.41451E-5			
R^2 = 9.53190E-1			
Sig = 0.00000E-1			
Correlation matrix			
1.0000	.8617	.7417	
.8617	1.0000	.9683	
.7417	.9683	1.0000	

SCREEN 3.3

Next let's try fitting the data to a user function. The x and y data sets will be defaulted to the X and R Variables that you selected earlier, so you can press Esc to go directly to the menu of fit options. This time execute User Function.

The screen will clear and a special window will open containing the text:

F(x) = _

Type in the fitting function as an algebraic expression. It should incorporate one or more A-I Parameters as fit coefficients. In this example we will fit the following trigonometric function:

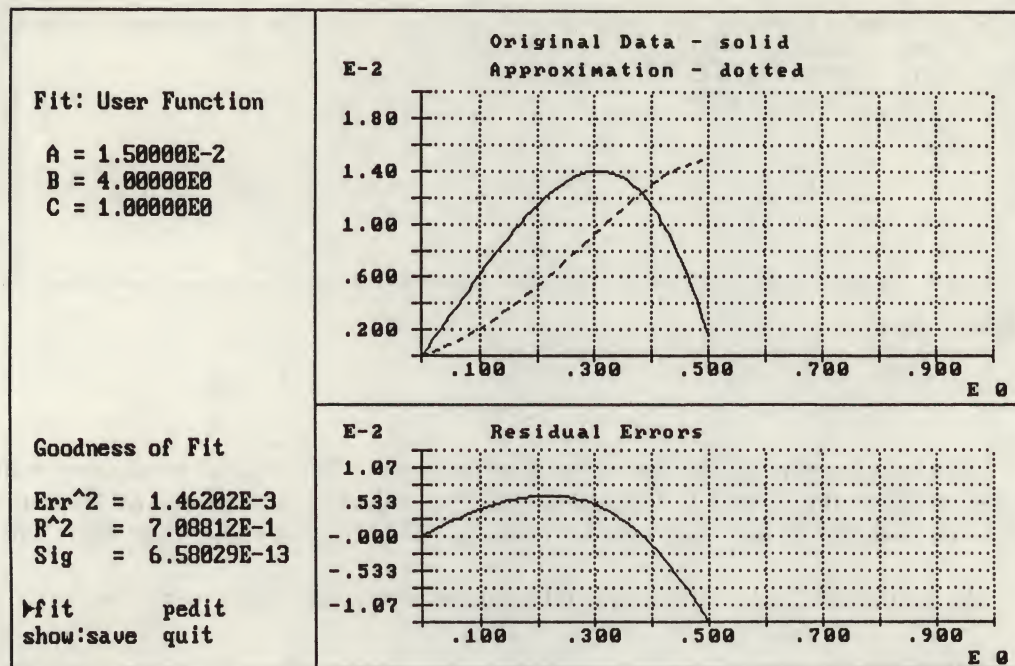
$$F(X) = A * \sin(B * X * X + C * X)$$

Now press the Esc key. The function will be compiled. If you made an error, an error message will be displayed and you will have the chance to edit the function. If compilation was successful, a long list of prompts will be displayed beneath the original window. These let you edit the contents of the fit Parameters, and to select Parameters to be fitted. Enter .015, 4.0, and 1.0 respectively for the values of the A, B, and C Parameters, and enter Y for each of the first three prompts in the second column to tell ASYSTANT to fit them.

Please note: The procedure used to fit user functions is fundamentally different from all the other fit options. With user functions, you must supply ASYSTANT an initial guess of the values to be fitted. The fitting routine then tries iteratively to improve on your estimates. The initial guess must be

reasonably close for the process to set off in the right direction. There are many subtleties to this type of curve fitting. Some of these points are discussed in Chapter 9. The other fit procedures in the Curve Fit Menu will directly calculate the best least-square fit without an initial guess.

The prompts at the bottom of the display allow you to make adjustments in the fitting process. Their significance is explained in Chapter 9. For this example we will stick with the default values, so press Esc to continue.



SCREEN 3.4

The screen will generate a display similar to the polynomial fit [refer to Screen 3.4]. However, this time the A, B, and C Parameters will be shown rather than the polynomial coefficients. The fitted function shown on this display is your initial guess, and has not yet been submitted to the fitting procedure.

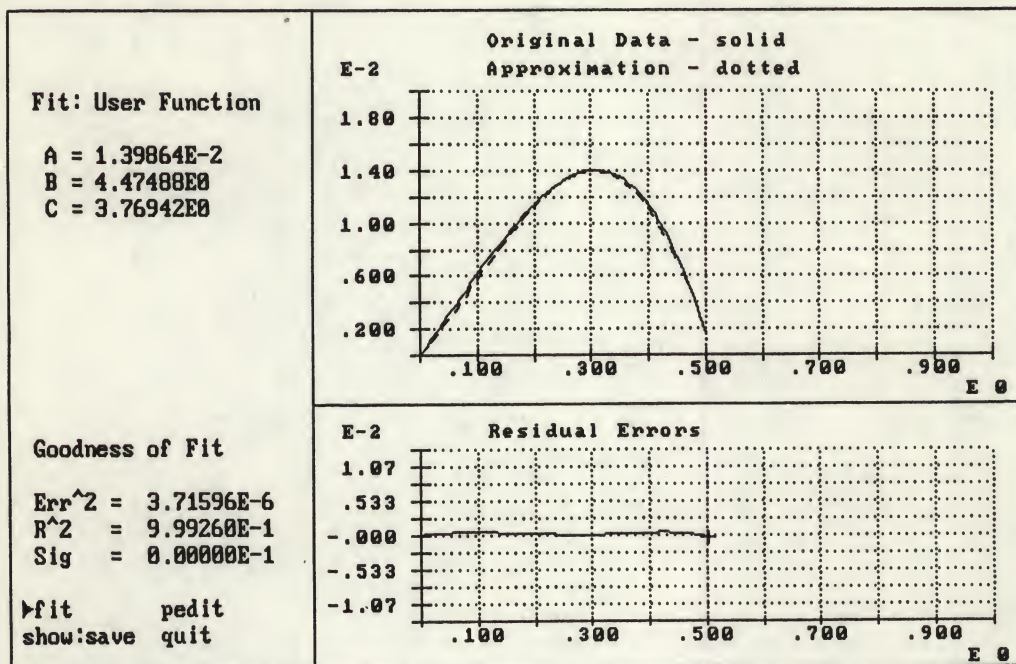
A menu at the bottom of the text window now provides the options:

fit pedit
show:save quit

The "quit" option will return you to the Main Menu. The "pedit" option lets you to edit the fit Parameters. You would select this if the plot showed that your

original estimates were grossly wrong, and you wished to improve them before starting up the automatic fitting routine.

The "fit" option invokes the fitting procedure. In this case, the estimates look good enough, so try executing fit four times. After each iteration, the Parameter values, the plot of the fitted curve, and the plot of the residuals are updated. Keep in mind that the data cannot be fitted precisely by this sine function, so you will find after executing fit three or four times that further executions of "fit" will show little or no improvement [refer to Screen 3.5].



SCREEN 3.5

Select show:save and enter a T by the "Save fitted values" prompt. Note the "Err^2" value shown on the screen: at approximately 3.7E-6 it is roughly an order of magnitude smaller than the "Err^2" of the quadratic fit. This indicates that the trigonometric function was indeed a better fit. We can also see this visually. Press Esc to return to the Variables prompt. R now contains our original data and S and T contain the quadratic and trigonometric fits, respectively. Esc and then Return to the Main Menu. Select Graphics. Use y Auto to plot R and then y Plot to overlay S and T. The T fit is clearly closer.

Return twice to get back to the Main Menu.

SESSION 4: **SIMULATING THE MOTION OF A CAR OVER SPEED BUMPS**

This session will introduce the use of the Wave Generator and some array processing capabilities of the Desk Calculator. The following functions will be used:

Wave Generator

to generate a square wave summed with random noise;

Desk Calculator

to generate a damped sinusoid;

to convolve two waveforms;

Graphics

to plot and display a resultant waveform.

The Wave Generator is designed to create periodic and noise waveforms useful in numerical simulations and as D/A output (with the ASYSTANT+) when a stimulus signal is needed for an experiment.

We will simulate the motion of a car wheel as it travels over speed bumps while approaching a toll booth. For readers who are not familiar with them, speed bumps on major highways are parallel strips laid on top of the pavement. When a car passes over them at high speeds, they generate enough noise and shock to encourage the driver to slow down. To make things more exciting, we will simulate a car that has bad shock absorbers, so that it will not take the bumps gracefully.

We will first generate the "speed bumps" with the Waveform Generator. Next, we will simulate the motion of the wheel in response to a single sharp bump. The "speed bump" waveform and the "wheel motion" waveforms will be convolved to illustrate the motion of the wheel as it moves across the bumps.

4.1 Generating the Waveform for the Speed Bumps

Select the **Wave Gen** option in the "Main Menu". Enter **500** into the prompt for the number of points in the waveform. The value of the random seed is not important in this example, so hit **Esc** (or **Return** twice) to accept the current value and continue.

▶Plot Wfrm	Clear M	M=M+0	Sine	Cosine	Square
Plot 0	Plot M	Plot M+0	Triang	Sawtooth	Pulse
Manip 0	Manip M	Return	Unif Noise	Nrml Noise	Pssn Pulse
No Data Manipulation		Plotting Out	Current Function: None		

Horz Specs	Vert Specs	Other Specs
#Pts/cy (strt):< 50.00 >	Gain:< 1.00 >	
#Pts/cy (end) :< 50.00 >	Bias:< .00 >	
	$O = G(I) + B$	

SCREEN 4.1

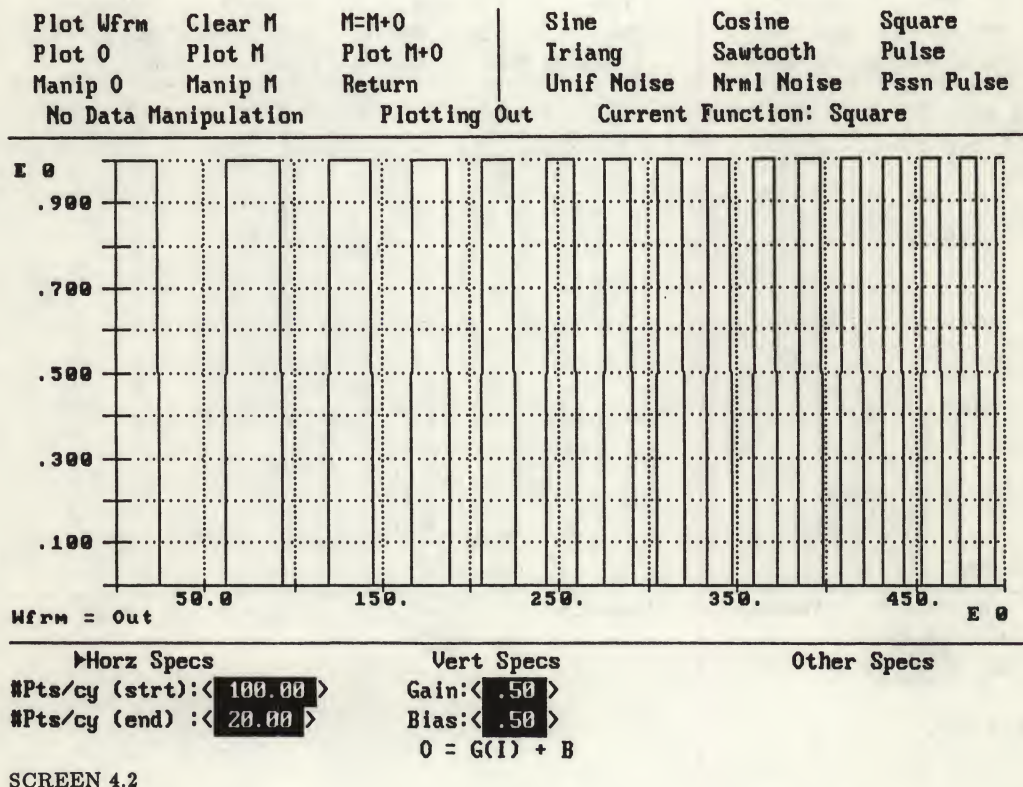
You are now inside the Wave Generator. The menu in the upper left corner is active, which is noted by the small arrow pointing to the first menu option "Plot Wfrm". [Refer to Screen 4.1]. Different types of waveforms can be generated from the Waveform Type menu, located in the upper right corner. Press **Tab** to switch between the two upper menus. (The "Tab" key is located on most keyboards above the "Ctrl" key on the left side of the keyboard. It is often marked with two arrows pointing in opposite directions). Waveforms are plotted in the middle of the display. The bottom portion of the screen lets you modify the horizontal and vertical attributes of the waveform using a combination of a menu and prompt lists. These function analogously to the frequency, gain, and bias knobs on a hardware function generator, but are more powerful. Press **PgDn** to enter the lower menu. Return to the top menus by pressing **PgUp**.

Enter the upper right menu, and select several waveform types in turn, eg., **Sine** and **Unif Noise**. As you press **Enter** to complete the selection, an **Output**

waveform will be generated and plotted. For now, you do not have to be concerned with the horizontal and vertical attributes, since reasonable defaults will be selected. Note the status line just above the plotting area changes as you generate different types of waveforms.

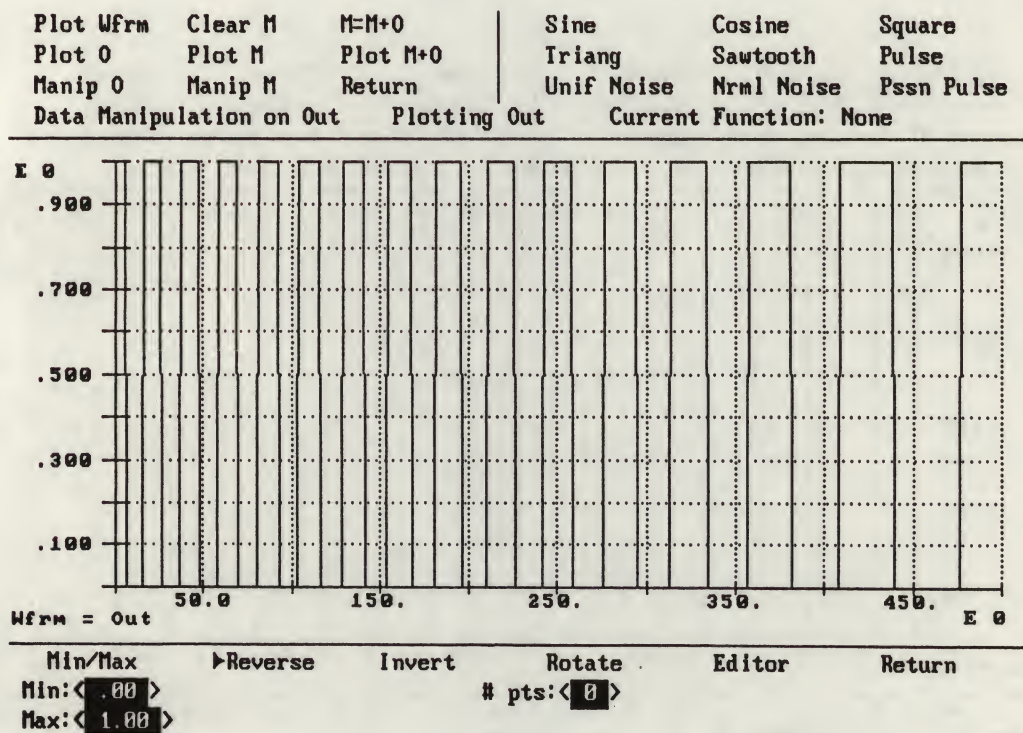
To continue with the example, we will model the speed bumps as the high portions of a square wave. Select the **Square** option (remember to hit **Enter**). The "bumps" created by the default vertical attributes span +1.0 to -1.0, which is unacceptable in our simulation since speed bumps are built above the road only, not below. We want to set the wave minimum to 0.0 and the maximum to 1.0 units. Hit **PgDn** to enter the Specs menu, and select **Vert Specs**. When you hit **Enter**, you will be in a prompt list allowing you to adjust the gain and bias of the Output waveform. Enter 0.5 for the gain prompt and 0.5 for the bias to give a minimum and maximum amplitude of 0.0 and 1.0, respectively. Hit **Esc** to accept the values you entered and exit the prompt list: the new waveform will be generated and plotted.

We are not yet done, however. Our simulation assumes the driver will react to the speed bumps by reducing speed. Thus, to realistically simulate the road displacement seen by the wheel as it travels over the bumps, the frequency of the square wave should decrease over time as the speed of the car decreases. We must introduce a frequency sweep into the square wave: it should start out at a high frequency (fast car speed) and end at a lower frequency (slower speed).



The frequency sweep is set with the "Horz Specs" option, located to the left of "Vert Specs" [refer to Screen 4.2]. Select **Horz Specs** (remember to press **Enter**). You will now be able to move the cursor around the prompt list. It lets you select the number of points per cycle at the beginning and end of the frequency sweep. Enter 100 for the starting points per cycle and 20 for the end. (You may notice that this is the incorrect order for what we want, but do it this way anyway). You must enter the "20" explicitly, because the "#Pts/cy (end)" automatically updates to 100 when you enter the starting value. Hit **Esc** to accept the values in the prompt list and exit to the menu. A new Output waveform containing the frequency sweep will now automatically be generated and plotted.

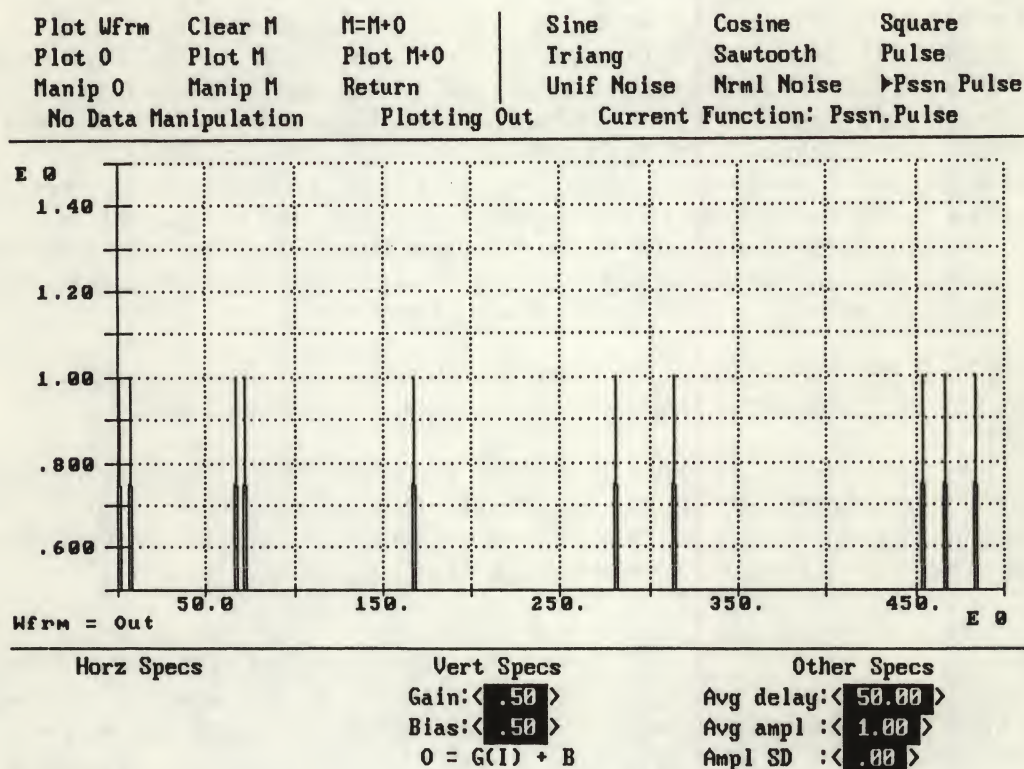
As already hinted, our output waveform is reversed: we want to have the high frequencies on the left (the car is going at high speed first). We could go back into "Horz Specs" and switch the start and end values for points per cycle, but instead let's illustrate using the "Manip O" option (for Manipulate Output) to reverse the waveform. Press **PgUp** and **Tab** to return to the top left hand menu. Select **Manip O** to enter the Manipulation submenu [refer to Screen 4.3]. Select the **Reverse** option. The **Output** waveform will now be plotted with the high frequencies on the left, as desired. Select **Return** to get back to the main menu.



SCREEN 4.3

Just for fun we can simulate some stones on the road by adding some impulsive noise to the square wave. The "M=M+O" option lets us create such a composite waveform in Memory. This option works much like a Memory+ button on a standard calculator. It takes the current Output waveform and adds it to the Memory waveform, clearing the Output wave when done. So, move to M=M+O option and execute it. Since M initializes to 0.0, we have just stored away our square wave. (Please note: the display continues to show the original square wave, even though "O" has been cleared. If you wish to confirm that "O" has indeed been cleared, select Plot O and then Plot Wfrm.)

Now let's create some impulsive noise using the "Pssn pulse" option in the Waveform Type menu. This will create randomly separated spikes following a Poisson Distribution. Press Tab to return to this menu and select Pssn Pulse.



SCREEN 4.4

You will see a set of impulses that are randomly spaced and have uniform height [refer to Screen 4.4]. Your display will not match the illustration exactly because a different seed to the random number generator will have been used. Note the prompt list beneath the "Other Specs" menu item in the lower right corner. Observe that we can change the average delay, average amplitude, and amplitude standard deviation (SD) of the impulses. We would like to adjust

these parameters to simulate different sized rocks (pebbles and stones), but we are ultimately concerned with how these impulses will add into the original square wave: the impulses cannot be too big, too small, or too frequent. Consequently, we would like to view the addition of the **Memory** and **Output** waveforms so that we can adjust the impulse parameters to our liking. That way we can preserve the current contents of **Memory** until we are sure we have an appropriate composite.

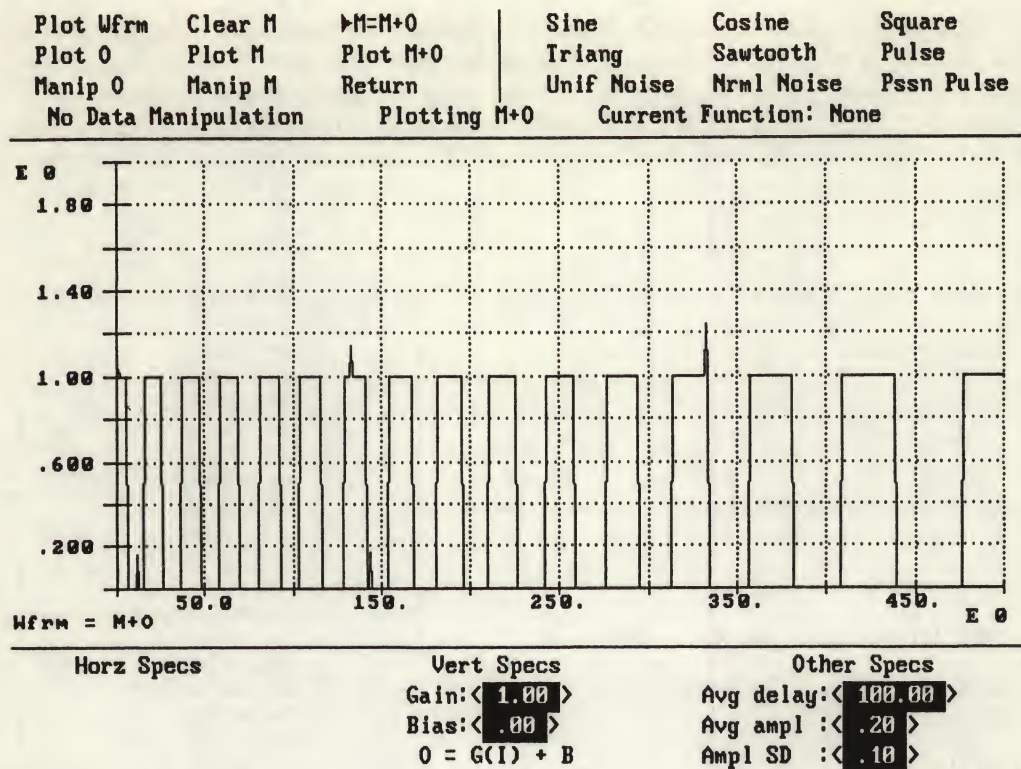
Press **Tab** to return to the upper left menu and select the **Plot M+O** option. On command, the Waveform Generator will now plot the sum of **Memory** and **Output**, without affecting their contents. (Note the change in the status line just above the plotting area.) Select **Plot Wfrm** to generate the composite plot.

Now press **PgDn** to move to the Specs menu on the bottom with the screen. Select **Vert Specs** and change the gain and bias to **1.0** and **0.0**, respectively, since we do not want to add a bias to the **Output** impulsive waveform. Remember, "Enter" lets you enter the prompt list, "Esc" lets you exit and generates the plot. Since you toggled the plotting parameter to "Plot M+O", you will see a plot of the addition of the **Memory** waveform and the adjusted impulsive waveform. Now select **Other Specs**. Change the average delay to **100** points. Also change the average amplitude to **0.5** and the amplitude standard deviation to **0.2**, simulating small rocks with a little variation in size. Now **Esc** to exit to see the addition of this impulsive waveform and the **Sum** waveform. Note that the impulses are still too large. Enter **Other Specs** again, and decrease the average amplitude to **0.2** and the amplitude standard deviation to **0.1**. **Esc** to see the new result, which should now be acceptable.

Press **PgUp** to return to the **Memory** contents (remember to press **Enter**). **Output** waveform to the **Memory** contents (remember to press **Enter**).

At this point we are ready to store our result in a Desk Calculator Variable. Wave Gen automatically prompts you to store results when you "Return" (we'll see this in a moment), but let's use the Mini Calculator here instead to illustrate some additional, powerful features of the Wave Generator.

Press the **Space** bar, and after a few moments the Mini Calculator will appear. Type **MEM** in response to the "Command:" prompt and hit **Enter**. An "Unnamed Array Real DIM[500]" will appear on the stack. This is the contents of the Wave Generator's **Memory**! You can now perform any operation the Mini Calculator is capable of. Afterwards, you can return the value on top of the stack using the command ">MEM" (read "to Mem"). One warning, however. Any array you wish to put back in **Memory** *must have the same dimension and data type* as the original **Memory** array. Trying to put another size array in **Memory** will generate an error.



SCREEN 4.5

In this case, we will simply store MEM in "S". Type **STORE S** Enter. Now that we are done, we can illustrate the use of ">MEM". Type **500 N:RAMP >MEM DONE** Enter. If you now select **Plot Wfrm** a diagonal line will be plotted. We've succeeded in transferring a calculator Variable to the Wave Generator. This is extremely useful if you have an existing array which you wish to manipulate (eg., to add noise for a simulation).

Now execute **Return** to get back to the Main Menu.

SCREEN 4.6

4.2 Creating the Wheel Motion with the Desk Calculator

In order to see how our car wheel will behave while travelling over the speed bumps, we must create a waveform that describes how the wheel will move over a single, sharp bump (the impulse response). We will suppose the motion is a damped sinusoid with a period of 30 points per cycle. We would also like the wheel motion to stop in about 4 periods (the shocks are pretty bad!). Thus, let's create a sinusoid 120 points long with a period of 30 points/cycle and multiply it by a damped exponential with $1/e$ duration of 30 points. First, however, we must make sure the system will evaluate the sinusoid using radians. Hit **PgUp** to enter the "Calculator Functions" menu and select **status** (first row, fourth column). Enter **R** for "Angular Units" (the fifth prompt down), and **Esc** back to the Main Menu.

Use either the command mode or the calculator menu, whichever you find easier, to create the waveform:

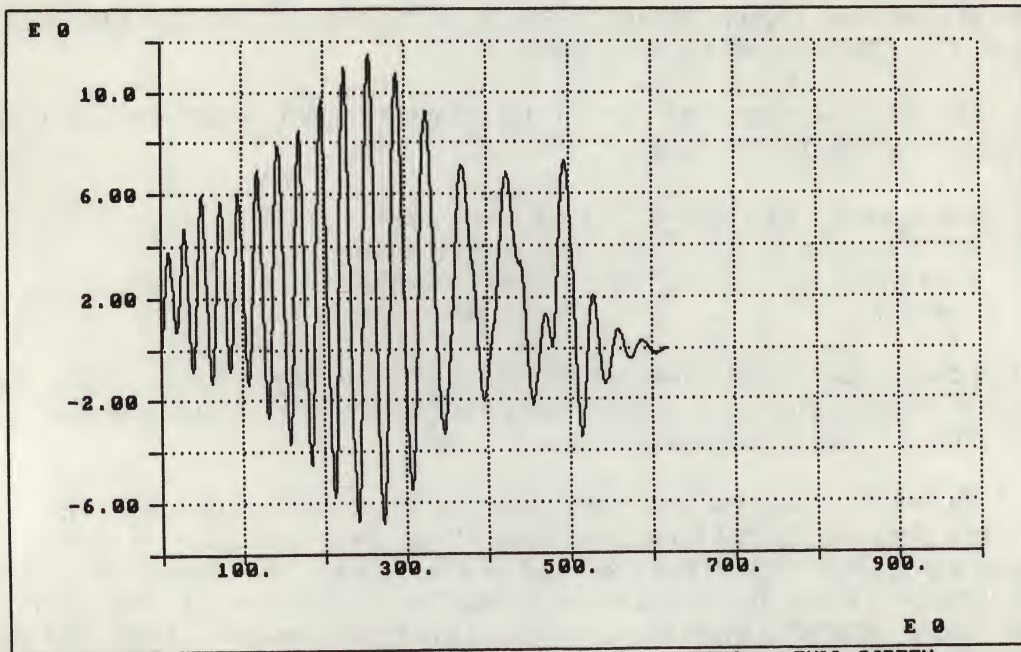
120 n:ramp 1 - 30. / store R	(to create a ramp)
R 2. * pi * sin	(to make the sinusoid)
R neg exp	(the damped exponential)
* store W	(store wheel motion in W)

(You could execute these lines one-by-one in the command mode to obtain the correct result.) You can plot the wheel response to a sharp bump with the "Graphics" menu, if you desire.

To help us remember what we're doing, select the **labels** option in the "Conversions and Special Functions" menu of the Desk Calculator. You will enter a prompt list which lets you type in a descriptive label for each Parameter or Variable. Move the cursor to the prompt for "S" and enter **BUMPS** and for "W" enter **WHEEL**. Press **Esc** to return to the Desk Calculator. Note that the labels appear to the left of the corresponding Variable listing on the Desk Calculator.

Now let's see what happens when the car goes over the speed bumps. To find the wheel motion in this instance, we must convolve the sharp-bump wheel response with the speed bumps. We can do this using the fourth Desk Calculator menu: hit **next** to bring up the "Wave and Matrix Operations" menu. Type **S W Enter** to put the bumps and the impulse wheel motion on the stack, and select **convolve**. This calculation will take a few seconds. When it is finished, type: **STORE T Enter**. We have completed the simulation of the wheel motion!

Now, to view the motion waveform, hit **PgDn** and select the **Graphics** option. Choose **y Auto** to plot "T". [Refer to Screen 4.7]. Remember this result shows the wheel motion as the car traverses the speed bumps. The larger the amplitude of this waveform, the larger the displacement of the wheel. Notice that the wheel motion starts to increase in amplitude, reaching a peak at about the 250-th point. The amplitude diminishes even before the car has left the speed bumps (at point 490), where it eventually dies to zero. This behavior is due to a resonance phenomena, and indicates that the car is sorely in need of new shocks!



Y Automatic Plot Window: FULL SCREEN
 y-array (R-Z): <I> 619 REAL
 ▶Display Graph To Plotter New Params Return

SCREEN 4.7

This ends Session 4. Select **Return** to return to the Desk Calculator.

SESSION 5: **ANALYSIS AND PROCESSING OF EKG WAVEFORMS**

This session will introduce the File Processor, Wave Processor and File I/O capabilities of ASYSTANT to analyze and process an EKG waveform (potentially acquired with ASYSTANT+). To begin, we will reduce noise in the waveform by averaging across multiple scans. Next, we will extract some statistics about the basic EKG itself and separate the high frequency components. Finally, we will save the results on disk. The following features will be used:

File Proc

*to average across multiple EKG scans on disk,
to transfer the result to a Variable;*

Wave Proc

*to interactively extract statistics from the data,
to smooth segments,
to extract the high frequency components;*

File I/O

to store the results in a disk file.

Please note: This session requires sample EKG data provided on your Key diskette in a data file **EKGDMO.DAT**. Please be sure this file is available before you continue. If you are working with a two-floppy system, you should copy it to any other DOS formatted diskette with enough room. You will insert it when needed in your B: drive. If you are working from a hard disk, the file should be in the same drive and subdirectory as the other ASYSTANT files. (If you are unsure whether you are using the correct disk, you can use the "Ctrl-D" option to scan for the **EKGDMO.DAT** file.)

5.1 File-based Processing

Let's assume that you have acquired an EKG trace and that you know that the signal is somewhat noisy. To compensate, you have wisely collected multiple scans of the EKG signal, each triggered off the same point in the heart cycle. This provides a reasonably repetitive signal that can be averaged across successive scans to reduce noise. The File Processor includes an option for averaging data in a disk file (among other file processing options). To use it, select **File Proc** from the Main Menu. The File Processor will then emerge [refer to Screen 5.1].

Dataset Location	I/O Status	
No dataset specified	Filename	# Subfiles
	Input: None	
	Output: None	
	Operation: None	
	Input var(s): R	
	Output var: T	
	Current DS: 0	

Main Menu	Variables
File > File	R= .0000
File > Var	S= .0000
Var > File	T= .0000
Return	U= .0000
	V= .0000
	W= .0000
	X= .0000
	Y= .0000
	Z= .0000

SCREEN 5.1

The windows in the File Processor provide you with continually updated information about the status of the current processing operation. As we work through this example, you will see the upper two windows updated to reflect your actions. We will explain this information as we go along.

Currently, the "Main Menu" in the lower left box is active. Choose **File > File**, since our input data is in a file, and since we also wish to save the results on file. (If we did not want to save the result on disk now, we could have chosen "File > Var", leaving the result in a Desk Calculator Variable only.) We can also start with a calculator Variable, process it, and store the result in a file ("Var > File"). Note, however, that at least one of the input and output datasets must be in a file.

5.1.1 SELECTING THE INPUT FILE

Selecting "File > File" displays the "Input File" screen. (Two disk floppy users should insert the diskette containing "EKGDMO.DAT" in the "B:" drive, if you haven't already). Enter the input file name, d:EKGDMO.DAT next to the File Name: prompt, where "d" is the drive containing the data file. When you hit return, ASYSTANT will read the file header and you will see a summary of the comments and subfiles in the file. (Subfiles are essentially Variables on disk. See Chapter 9 for further information.) You can scan the contents of as many data files as you wish simply by pressing the "left arrow" and entering new file names. (If an error message occurs, you have either typed the file name incorrectly, specified the wrong drive, or inserted the wrong disk. Press Ctrl D, if necessary, to call for DOS Operations to locate the file.)

Once the filename is properly entered, press C (or hit Enter) to continue. A new prompt list appears to let you select which portions of the data to process [refer to Screen 5.2]. To avoid errors in the reading of data on disk, some restrictions have been placed on the regions from which you can grab data. Selecting these regions requires an understanding of the organization of ASYSTANT data files.

Input File			
File Name: (C:EKGDMO.DAT)		Quit/Continue: * C *	
8 Comments		Subfiles (Total #: 10)	
1)	Start#	Shape	#Repts
2)	1	1 x 400	10
3)			
4)			
5)			
6)			
7)			
8)			
Subfile: (1) Row (0=all): (0) Start column: (1) 1 x 400 (Max Length: 4000) # of columns: (400) Offset (columns): (400) # datasets: (10) Scroll/Plot/Edit/List/New file/Quit/Continue: * C *			

SCREEN 5.2

Each subfile in an ASYSTANT data file is the "image" of a calculator Variable. The number of rows and columns in a subfile are measures of the size of the first and second dimensions of the original Variable. In the EKGDMO.DAT file there are 10 subfiles, each of which is a 400 point EKG scan, for a total of 4000 points. Many data files, including this one, have sequences of multiple, identically shaped subfiles. ASYSTANT calls such a sequence a "block" and allows you to access data only within the bounds of a block. Thus, you cannot grab data that spans two different shaped subfiles. To strengthen the concept of a block of subfiles, ASYSTANT provides information about the current block in most prompts dealing with file I/O. In this example, you can see that the block is 4000 points long, as indicated by the text **Max Length: 4000** to the right of the prompts.

Note that the prompt list was automatically filled in. The current subfile is 1, and the number of columns is 400, reflecting the size of subfile 1. If our data had more than one row, we could specify the location of the data by row as well.

Remember that we want to average across all 10 EKG scans to produce a single, less noisy composite. To do this, we should average the data reasonably near the beginning of each scan, since data closest to the trigger point will exhibit the least amount of cyclic jitter. Thus, we will average only the first 250 points in each subfile.

The ASYSTANT allows you to choose arbitrary regions of data in the current block by selecting a subfile, a start column relative to the beginning of that subfile, and a number of columns. Such data segments are termed "datasets". In our example, we want to average multiple datasets, each originating from the first 250 points of successive scans. It is important to note that a dataset can span multiple subfiles within the current block. In this instance a single dataset could include as many as 4,000 points. However, please keep in mind that a dataset must be stored in your computer's memory when it is actually processed. Therefore, ASYSTANT may run out of memory during processing if you select too large of a dataset. In this case it will give you an error message. The maximum number of data points ASYSTANT can handle in a single dataset will depend on the dataset's type (single or double precision; integer, real, or complex) and the processing operation selected. If you do get an error, you need to divide your input data file into smaller individual datasets.

In this example, each separate scan is in a single subfile. Therefore, each dataset in this example will also be contained within a single subfile.

Input File			
File Name: <C:EKGDMO.DAT>		Quit/Continue: * C *	
8 Comments		Subfiles (Total #: 10)	
1>	Start#	Shape	#Repts
2>	1	1 x 400	10
3>			
4>			
5>			
6>			
7>			
8>			
Subfile: < 1 > Row (0=all): < 0 > Start column: < 1 > 1 x 400 (Max Length: 4000) # of columns: < 250 > Offset (columns): < 400 > # datasets: < 10 > Scroll/Plot/Edit/List/New file/Quit/Continue: * C *			

SCREEN 5.3

Let's proceed with the dataset specification [refer to Screen 5.3]. We want to process from scan 1 to the last scan (i.e., 10). The first dataset is located in subfile 1, which is already selected. There is only 1 row in these scans, so a "0" for all rows or a "1" is equivalent for the **Rows (0=all):** prompt. The beginning of the dataset will be column 1, which is already selected. However, we want the number of columns in the dataset to be 250 and presently the **# of columns:** prompt shows 400 columns. Enter 250 to set the proper dataset column size.

Notice that when you entered this number, the next two prompt values changed. ASYSTANT automatically selects values so that as many datasets as possible span the current block of subfiles. ASYSTANT is trying to give you all of the data in the block; note that it now specifies 16 datasets ($16 \times 250 = 4000$, the total number of points in the block). We do not want all possible datasets, however, only those that start at the beginning of each scan. Since each scan is 400 points, the beginning of each subsequent dataset must be separated by 400 points. We need to change the **Offset (columns):** prompt, which specifies the column offset from the start of one dataset to the next. Currently, you should see an offset of 250. Instead, enter 400, and again notice how the prompt values change. You will see that the **# datasets:** prompt now indicates 10, which is the number of datasets that we want. Consequently, skip this prompt, and go on to the next one.

The last prompt offers the following choices:

- Scroll - to graphically scroll each row of a block of subfiles onto the screen (this option is particularly useful in selecting datasets)
- Plot - to plot the current dataset on the display
- Edit - to edit file comments
- File - to display a list of all subfiles
- New file - to select a new input file
- Quit - to return to the Data Processing menu
- Continue - to continue to the Output File menu

First let's plot the currently selected dataset. As you can see from the prompt list, the current dataset is located in subfile 1, row 1, columns 1 through 250. Enter **P** to see a plot of the dataset.

Press **Esc** after examining the plot to move back to the previous prompt. Now enter **C** to specify the output file.

5.1.2 SELECTING THE OUTPUT FILE

You may write the selected datasets to a new file or append them to any existing file, including the input file. If the input file is chosen, you also have the option to write the data back to the same location from which it was read (this is called writing "inplace"). Since writing inplace destroys your original dataset, use it only when necessary, and only after backing up your original data. In this case you wish to append the processed datasets to your original input file, so enter the name **d:EKGDMO.DAT** as the output file name. If you mistype this name, **ASYSTANT** will think you wish to specify a new file. You'll see a prompt "New File" and the "Comments" and "Subfiles" frames will be empty. If this happens, simply hit **Left Arrow** and retype the file name. When you are sure that "**d:EKGDMO.DAT**" is properly specified as the output file, press **C** to continue.

The final prompt offers you the following choices:

- Append - to select append as the output method
- Inplace - to select writing inplace as the output method (only available if input file=output file)
- Edit - to edit file comments
- File - to display a list of all subfiles
- New file - to select a new output file
- Quit - to return to the File Proc Main menu *without writing the data to the disk output file*
- Continue - to continue specifying processing operations

The default output method is Append, so enter **C** to append the data to **EKGDMO.DAT** and continue setting up processing. Note that no data is actually written until processing starts.

5.1.3 SELECTING THE PROCESSING OPERATION

Dataset Location		I/O Status	
Start subfile:	1	Filename	# Subfiles
Row # (0 = all):	0	Input: C:EKGDMO.DAT	10
Start column:	1	Output: C:EKGDMO.DAT	10
# columns:	250	Operation: DS.Average	Unary
Offset (columns):	400	Input var(s): R	
# datasets:	10	Output var: T	Append
		Current DS: 10	

Unary ops		Binary ops	
Abs	Sqrt	+	-
Log	10^	*	/
Poly Scale	User function	Max	Min
FFT	IFFT	Catenate	Laminate
Pwr spec	Autocorr	Convolve	Crosscorr
Smooth	DS Cat & Smooth	Quit	
DS Average	DS Stndrd Dev		
DS Subsection	DS Reshape		

SCREEN 5.4

You should now see a menu of processing options [refer to Screen 5.4]. Note that the "Dataset Location" and "I/O Status" windows on top of the screen now reflect the specifications we've already completed. What remains now is to select an operation and input and output Variables (see also 5.1.4 below). These Variables act as buffers during processing. Datasets are read one at a time into the input Variable, the processing operation is performed, and results left in the output Variable. If file output is specified, the result is then written to the file.

The processing options menu is divided into two separate menus. "Unary Ops" require only one input Variable -- in this case read in one dataset at a time from the input file. "Binary Ops" require two input Variables: one usually from an input file, the other always from a Desk Calculator Variable.

You can switch between the "Unary Ops" and "Binary Ops" menus using the "Tab" key. Hit Tab twice to toggle back and forth. You may wish to study both menus to become familiar with the file processing options provided. Reference Chapter 8 provides complete descriptions of each option. The operation we want is **DS Average** in the "Unary Ops" menu. Note that the I/O Status window updates to show the operation selected.

5.1.4 SELECTING THE INPUT AND OUTPUT VARIABLES

Dataset Location		I/O Status	
Start subfile:	1	Filename	# Subfiles
Row # (0 = all):	0	Input: C:EKGDMO.DAT	10
Start column:	1	Output: C:EKGDMO.DAT	10
# columns:	250	Operation: DS.Average	Unary
Offset (columns):	400	Input var(s): U	
# datasets:	10	Output var: V	Append
		Current DS: 0	

I/O variable selection for unary ops	Variables
Input variable (R-Z): (U)	R= .0000
Output variable (R-Z): (V)	S= .0000
Quit/Continue: * C *	T= .0000
	U= .0000
	V= .0000
	W= .0000
	X= .0000
	Y= .0000
	Z= .0000
{operation(input var)} = {output var}	

SCREEN 5.5

As already noted, it is necessary to specify input and output Variables to act as buffers during processing [refer to Screen 5.5]. Since we selected a unary processing operation, we are prompted for only a single input Variable. Data will automatically be pulled from disk into the specified input Variable during processing. The result will be stored in the output Variable, and then written to disk at the end of processing. **Note:** the contents of the selected input and output Variables will be overwritten by the processing procedure. Therefore, always select unused or unneeded Variables at this stage. Choose U as the input Variable and V as the output Variable. Notice that the input and output Variables are indicated in the upper right hand window as you make your selections.

5.1.5 PROCESSING AT LAST

Press **C** to start processing. A message will indicate that processing has begun. You should see activity on the disk drive containing the "EKGDMO.DAT" file. In the upper right corner the value beside the "Current DS" label increments by one as each new dataset begins processing. You can interrupt the processing of your data by hitting "Esc" (don't do it). Were you to press the "Esc" key, processing on the current dataset would continue to completion, but no additional datasets would be processed. This approach ensures that the File Processor will recover in a "clean" state.

After a few moments the screen will be repainted with the File Proc Main Menu in the bottom left corner. In the "I/O Status" window you will notice that the number of output subfiles is one larger than the input subfiles. This confirms that the average has been appended to the output file. You will also notice that Variables "U" and "V" contain 250 point arrays. "U" contains the last dataset read from the input file, and "V" the last output dataset, i.e., the average of the datasets in this example. We can therefore pull the average from "V" directly, rather than read it from the dataset last appended to the disk file.

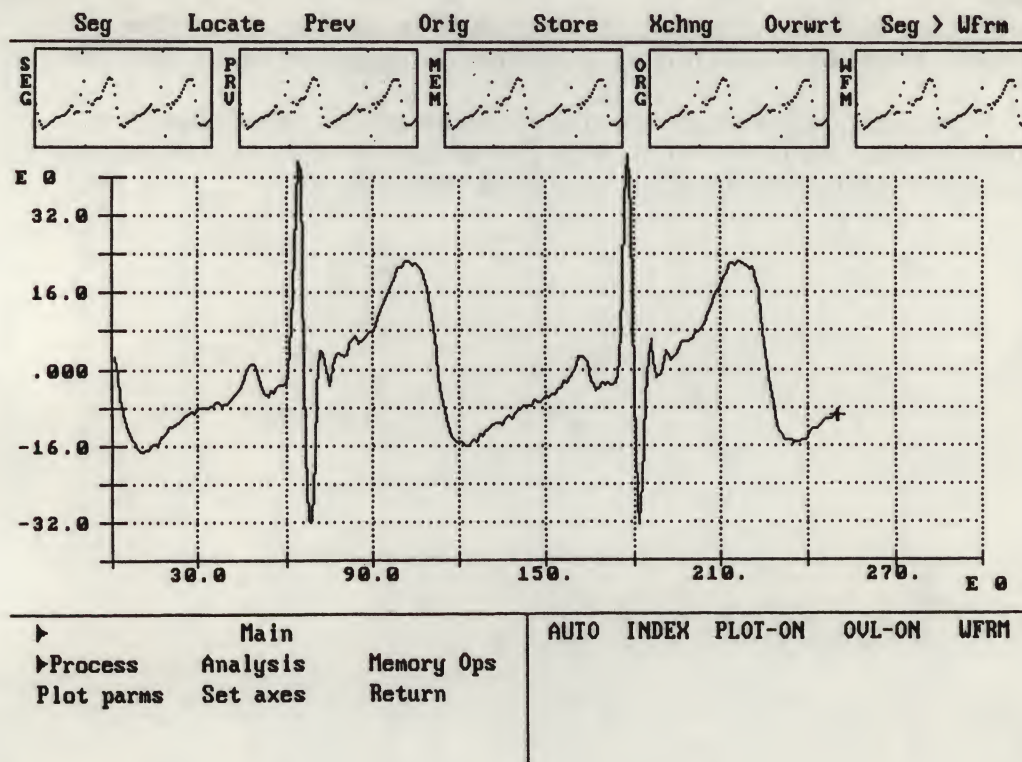
At present, we are finished with the File Processor. Select **Return** to re-enter the Desk Calculator. In the next part of this session, we will use the Wave Processor to analyze the data just processed from disk.

5.2 Waveform Processing Analysis

In this section, we will extract the high frequency component of the EKG signal and do some simple analysis on it. Since we are not sure of the ideal parameters to use in processing, it would be useful to see results after each step. The Waveform Processor enables us to do this.

Select **Wave Proc** in the Desk Calculator Main Menu and then enter **V** as the Variable to process. (Notice that the prompt defaults to the first Variable containing an array. Make sure you replace the default value with **V**). Now enter **P** to process **V**.

5.2.1 WAVEFORM PROCESSOR LAYOUT



SCREEN 5.6

The Waveform Processor screen is divided into a series of windows [refer to Screen 5.6]. The large central window is used to graphically select regions of the input array for processing, to plot the results of processing, and to graphically choose regions of the data array for analysis.

Just above the plotting region are a set of five small windows. Each window represents a memory or repository. Right now, all of the windows are

identical. As you process different segments of the waveform, they will be updated to contain different stages of processing. More details will be provided later in this session.

At the top line of the screen is the Segment menu, which is used for controlling the contents of the five repositories. At the bottom left corner is the Wave Processor Main menu. Each option on this menu provides access to a submenu. In the lower right, the plotting options currently in effect are listed.

5.2.2 PLOTTING THE DATA

Since the Wave Processor is graphically oriented, several plotting options are available from the "Plot parms" and "Set axis" submenus. "Plot parms" allows you to overlay the contents of different repositories. "Set axes" is used to set the horizontal and vertical axes ranges and units. Choices made in these two submenus are reflected in the line of text directly below the plot in the lower right corner. We use several plotting options in the course of this session. For now, select the Set axes menu to set the range of the horizontal axis.

First, take a look at the center plot. Notice that the units on the horizontal axis are the array indices: the last point is plotted at 250. Sometimes index units are useful, but generally the horizontal axis of a waveform will correspond to "real-world" units (usually time). The current waveform input from "V" has units of seconds, so we will change the axis units.

"Set horiz" lets you set the horizontal units and range, so that you can view your data in a meaningful "world". In this example, the EKG signal was collected at a sampling rate of 100 Hz. Thus the 250 points in the input waveform span 2.49 seconds. To set this range, select Set horiz, which is an abbreviation for "set the horizontal world". When a prompt appears at the bottom of the screen enter 0 for the minimum horizontal unit and 2.49 for the maximum. You also have the option to enter a unit label. The label default is "sec", which is appropriate in this case, so you need not enter a label. When all entries are acceptable, press Esc.

By itself, the "Set horiz" will have no apparent effect on the Wave Processor. Unless you tell it otherwise, the Wave Processor will continue to plot in index coordinates. So, select Indx/Wrld. Note that the current plotting status changes from "INDEX" to "WORLD". Each time you select this option, the plotting status will toggle. If left in the "World" toggle, the Wave Processor will now plot with units of seconds on the horizontal axis. Select Plot data to replot the waveform.

The use of indices or real-world axis values extends to all references to the horizontal axis. Thus processing or analysis results will be reported in different units, depending on the units entered with "Set horiz" and the "Indx/Wrld" toggle.

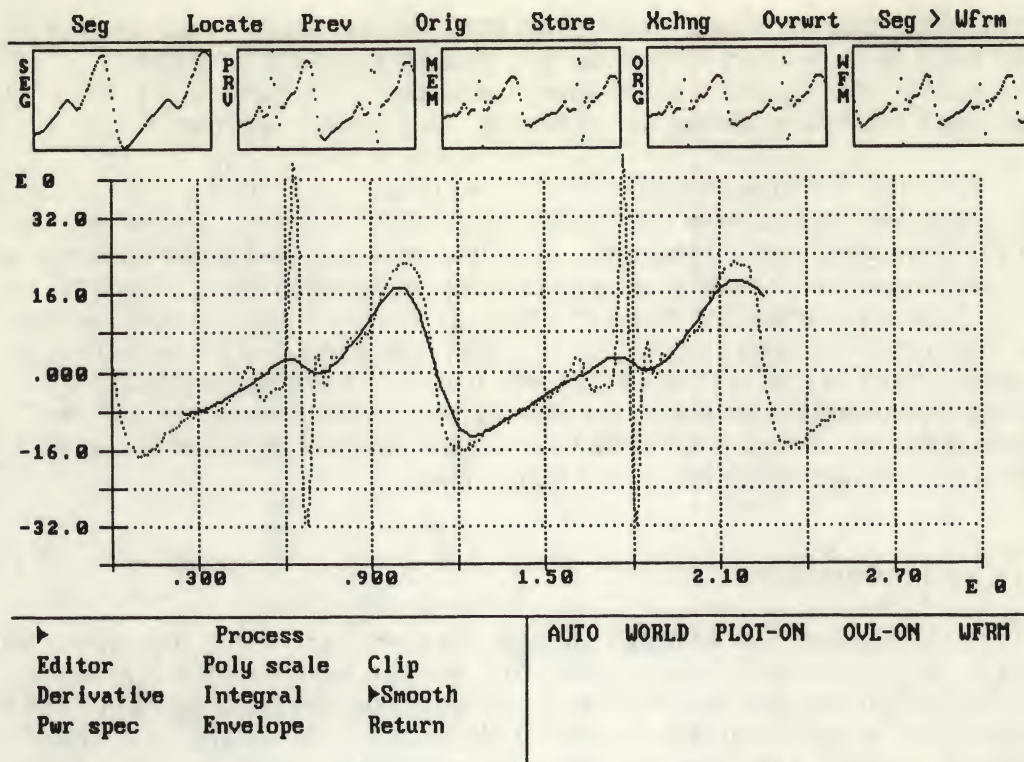
5.2.3 PROCESSING A SEGMENT OF THE ENTIRE WAVEFORM

We will limit processing to the central portion of the waveform, since we are interested only in the high frequency component. We need to tell the Wave Processor explicitly what segment of the overall waveform we want.

Press **PgUp** to go to the Segment menu at the top of the screen. Now select **Locate**. A prompt will appear at the bottom of the screen asking if you want to select the segment numerically or graphically. We know roughly the location of the data segment we want so choose **N**. A new prompt list will now appear asking for the desired left and right endpoints of the segment. Notice that the current values are in real-world units; if "INDEX" mode were in force, these numbers would be listed as indices. Enter **0.25** for the left bound and **2.25** for the right bound. Press **Esc** to accept the new values, and you will notice a change in the leftmost four memory plots at the top of the screen, indicating that these repositories now contain the segment just selected. Notice that the rightmost plot, labelled **WFM**, still contains the complete, original waveform. This illustrates its function: to remember the original waveform. If we make a mistake in subsequent processing, we can always recover the original waveform from this repository using "Locate" to select a segment or the whole waveform again.

Had we chosen to graphically locate the segment, **ASYSTANT** would have opened a waveform scroller and marker which would allow you to scroll across your waveform and mark the left and right bounds of the region that you wish to process. This is an easy, interactive means of marking a segment and, in many instances, is better than numerical location. After this session is completed, we suggest you go back to the "Locate" option and try selecting another segment graphically. You will need to refer to Chapter 6 of the Reference section.

Press **PgDn** to return to the bottom left menu. Note that the submenu you were using is still current, not the Main menu. Execute **Return** to get back to the Wave Processor Main menu. Now select **Process**. [Refer to Screen 5.7]



SCREEN 5.7

The Process submenu contains waveform processing options. Remember our goal is to extract the high frequency component of the waveform. We will first smooth the waveform to obtain its low frequency component and then subtract the original waveform, leaving only the high frequencies.

Select **Smooth**. A prompt will ask for the cutoff frequency in cycles per point. (**Smooth** uses a Blackman window with a low pass filter -- see Reference Chapter 6 for more details). Enter **0.3** and press **Esc**. After a few seconds, you will see the leftmost two repository plots update and a waveform plot in the center of the screen. Notice that the smoothed segment between 0.25 and 2.25 seconds is overlaid on the main waveform plot, and the leftmost repository labelled **SEG** (for current segment) now contains the smoothed segment. The contents of the **SEG** repository are the input values for any processing operation. On completion of the operation, **SEG** is then updated to contain the results. The **PRV** (for previous) memory, just to the right, is also updated each time a processing operation is invoked: **PRV** holds the previous value of **SEG**, preserving the input values, while **SEG** is updated by the processing step.

The result we obtained still has quite a few high frequencies in it; we must lower the cutoff frequency. Let's restore the previous (unsmoothed) segment and try again with a lower cutoff frequency. Press **PgUp** to move to the Segment menu and select the **Prev** option. This will copy the contents of **PRV** back into

the current segment (SEG). Now press **PgDn** key to return to the lower menu, and select **Smooth** again. This time use a cutoff frequency of **0.075**. (Remember to press **Esc** after entering the value.) Now you will see a heavily smoothed waveform plotted on the screen. This result is acceptable.

To obtain the high frequency component of the segment, we must subtract this smoothed version from the original unsmoothed segment. To retrieve the original segment, we could save the smoothed segment, then restore the original waveform and use **Locate** to grab the segment as we did before. However, at this point a much simpler means is available. Notice that the original segment is still in the repository labeled "MEM". This repository holds a segment in "memory" and can be used much like the memory on a hand calculator. A special submenu is available to use this segment, and is entered through the Main menu by selecting "Memory Ops". Select **Return** to go back to the Wave Processor Main menu and execute **Memory Ops**.

5.2.4 MEMORY OPERATIONS

In the Memory Ops submenu you will see several arithmetic operations that can be performed on SEG and MEM. You can also store the SEG contents in MEM or exchange the two. We want to subtract the smoothed segment from the original, so we must exchange SEG and MEM before we use the "Seg - Mem" option. Execute **Exchange**, and watch the contents switch in the "SEG" and "MEM" plots. Next, select **Seg - Mem**. A prompt list will open allowing you to select scale factors (multiplying constants) to apply to the individual segments before carrying out the arithmetic operation. Since we want to subtract the data just as it is, we want scaling factors of 1.0 for both prompts. This is the default, so press **Esc** to accept the prompt list and continue. The result of any arithmetic operation is placed in the current segment repository. You will see the "SEG" plot update, and the plotting area will display an overlay of the original waveform and its high frequency component.

Note that the Mini Calculator commands "MEM" and ">MEM" described in Session 4 with the Wave Generator also work with the MEM segment of the Wave Processor. Simply hit the space bar to bring up the calculator, and type "MEM" to put the contents on the stack. Conversely, ">MEM" puts the contents of the top element of the stack into the MEM segment. Again, use >MEM only with arrays identical in size and type to the current segments.

For easy comparison, it would be interesting to view the high frequency component (in SEG) and low frequency component (in MEM) in a single plot. We can do this from the PlotParms submenu. Execute **Return** to get back to the Wave Processor Main Menu, and select **PlotParms**.

Select **Mem ovl**. Note the change in the plotting status line in the lower right of the screen. It now indicates that "MEM" will be overlaid with the current segment. Now select **Plot data** to see the overlay. (For fun, select **Ovl tgl** and execute **Plot data** again. Note the memory segment is no longer plotted and the plotting status line indicates "OVL-OFF". Select **Ovl tgl** again to return to the "OVL-ON" state and replot the data with **Plot data**.)

Before analyzing the high frequency component, let's save this data in one of the Desk Calculator Variables. Return to the Wave Processor Main menu with **Return**, and select **Return** again, this time in the Main menu. A prompt list will appear [refer to Screen 5.8], letting you select the Variables in which to save data. We want to save only the current segment contents, so enter **W** in the **Current segment:** prompt. "**D**" defaults in the next two prompts, which is what we want, so move to the **Save/Quit&save/Process** prompt and enter **S**. You will see the "**W**" variable update to reflect its new contents. Now enter **P** (for process) to return to the Waveform Processor, so we can finish the analysis of the high frequency component. Note that when you return, everything appears as when you left it.

Stack Contents	Parameters
(empty stack)	A= .0000 B= .0000 C= .0000 D= .0000 E= .0000 F= .0000 G= .0000 H= .0000 I= .0000
Wave Proc Output Selection	Variables
Save in R,S,T,U,V,W,X,Y,Z,# (stack) or Delete	R= .0000 S= .0000 T= .0000 U= 250 REAL V= 250 REAL W= .0000 X= .0000 Y= .0000 Z= .0000
Current Segment : (W)	
Memory Segment : (D)	
Current Waveform: (D)	
Save/Quit&save/Process: * Q *	

SCREEN 5.8

5.2.5 ANALYSIS OF THE SEGMENT

Select **Analysis**. The Analysis submenu has some routines commonly used for analyzing data. All of the items except "Note #'s" and "Return" will replot the data and turn on graphics cursors. The cursors let you visually select regions of the segment you want to analyze. First, we will extract some statistics about the two high frequency blips (the large amplitude portions of the waveform).

Select the **Stats** option. You will see two vertical lines (cursors) that can be moved to the left and right with the arrow keys. Motion is limited to the region spanned by the current segment. Press the **End** key. Now the left and right arrows control the left cursor. Press the **PgDn** key. The arrows control only the right cursor. Press the **Down** arrow key. Now both cursors move together.

Move the cursors as necessary, so that they surround the leftmost blip. You may find that the cursor movement is too coarse to accurately surround the blip. Hit **PgUp** followed by the number **1**. You can now move the cursors in a fine increment. You can change to a larger increment using the same procedure: just enter a larger integer following **PgUp**. Try entering **20**.

Press the **Ins** key. You will see numerical results appear in the lower right corner of the screen and some lines appear in the plotting area. Three horizontal lines can be seen: the middle indicates the mean of the data within the cursors, while the outer two show the mean, plus and minus one standard deviation. Now move the cursors so that they surround the rightmost blip, and press the **Ins** key. The analysis will again appear in the lower right. All of the analysis options share this kind of interactive data selection and reporting.

Press **Esc** to return to the Analysis submenu and select the **Note #'s** option. This will output the results currently printed in the lower right window to either the printer or the current Note file. If you select the Note file, you can choose to append the data to the end of the file or insert it at the current text cursor position. Let's output to the end of the Note file. Enter **N**, for Notepad, **C**, for Continue, and then **E** for end of file. Once the numbers have been written into the file, you will be returned to the Analysis submenu. You may verify that the numbers are at the end of the Note file by using the **Ctrl-N** option.

We have finished analyzing and processing the average EKG dataset! All that remains is to save the processed waveform on Disk. Choose **Return**, and then **Return** again to exit the Wave Processor. Choose **D** (for delete) for all three save options on the exit prompt list, since we already saved the desired data. Then select **Quit&save** to return to the Desk Calculator.

5.3 Transferring Variables to disk

Remember that we saved the high frequency component of the EKG waveform in the "W" Variable. Let's use the File Input/Output menu to save this data on disk. This option lets you easily move data between disk files and

Desk Calculator Variables. The data transfer portion of this menu is very similar to the "Input File" and "Output File" prompt lists you've already seen in the File Processor.

Select **File I/O**, and you will see the lower left portion of the screen replaced by the File Input/Output Main menu. Besides "File/Var Transfer", this menu gives you the ability to convert files from other software so they can be read by ASYSTANT and vice versa. This is covered in Appendix A of this tutorial. We want to save the contents of **W** in an ASYSTANT file, so select **File/Var Transfer**. The screen will be replaced by one that you have seen in the File Processor menu. Let's choose the "d:EKGDMO.DAT" file as the output file; we'll append the data in **W** to the end of the file. Select Continue after the comment and subfile summary has been printed.

The bottom part of the screen will be replaced by a set of prompts. Many should be familiar now, since you saw them in the File Processor. [Refer to Screen 5.9] Enter **W** in the **I/O Variable** prompt. The next prompts: **Subfile:**, **Row (0=all):**, **Start Column:**, and **# of Columns:** are used to locate the dataset to be read from or written to the disk file, as you will recognize from your experience with the File Processor. (Chapter 9 of the Reference Section details using these prompts). Since we want to append "W" to the end of the file, we do not need to change any of them. Move to the last prompt.

File Input/Output			
File Name: <C:EKGDMO.DAT>		Quit/Continue: * C *	
8 Comments		Subfiles (Total #: 11)	
1)	Start#	Shape	#Repts
2)	1	1 x 400	10
3)	11	1 x 250	1
4)			
5)			
6)			
7)			
8)			
I/O Variable (R - Z): < W >		201 REAL	
Subfile: < 1 >		1 x 400 (Max Length: 4000)	
Row (0=all): < 0 >			
Start Column: < 1 >			
# of Columns: < 400 >			
Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit: * Q *			

SCREEN 5.9

The last prompt offers the following choices:

- Read** - to read data from a dataset file into a Variable
- Write** - to write data from a Variable into a dataset file
- Append** - to append data in a Variable to the end of a file
- Scroll** - to scroll each row of a block of subfiles onto the screen (this option is particularly useful in selecting datasets)
- Plot** - to plot the current dataset on the display
- Edit** - to edit file comments
- File** - to display a list of all subfiles
- New File** - to select a new input file
- Quit** - to return to the File I/O Main menu

Since we want to append the data on the end of this file, choose **Append**. Once the data has been appended, select **Edit** to update the comments to reflect the appending of the average EKG dataset and the high frequency component of this average dataset. After entering comments, hit **Esc** to get back to the action prompt. Next, choose **Quit**, return to the File I/O Main menu, and select **Return** to go back to the Desk Calculator.

SESSION 6: **A STATISTICAL ANALYSIS OF STOCK PORTFOLIOS**

In this Session, we will attempt to analyze some of the factors that may influence the share price of corporate stocks. This analysis will begin with some of the basic features of the Statistics menu and lead to some of the more advanced analysis capabilities, such as ANOVA and stepwise regression. Features covered include:

File I/O (optional)

to convert an ASCII data file to ASYSTANT format;

Statistics

to test hypotheses,

to perform a table ANOVA,

to perform a regression analysis.

Needed for this session (optional):

the file STOCK.ASC (on the Analysis Overlay Disk),
Appendix A.

The hypothesis testing and ANOVA will be used to analyze some financial information from several companies with the goal of narrowing consideration to a single company. The regression analysis will then be used to provide a prediction of the share price of this company after a year. Since we have an opportunity for a risk-free investment at 8% (e.g., a savings bond), the predicted price must exceed the current price by 8% if we are to purchase it. Please note that these analyses do not conform to current theories of stock valuation and do not represent a reasonable method for selecting investments.

Two groups of numbers will be used. The numbers in the initial group were taken from the January 13, 1986 issue of Forbes, while numbers for the second group were extracted from a corporate annual report.

6.1 Choosing the Company -- Hypothesis Testing and ANOVA

6.1.1 BACKGROUND

The first question one might ask before choosing a company in which to invest is: what factors are most important in determining its attractiveness? Three widely published growth factors are: return on equity (ROE), earnings per share (EPS), and market sector.

Earnings per share is the ratio of net earnings to the number of shares outstanding. It is a measure of the return per share, and is unaffected by the market price of each share. Unlike the other two indicators, it is a dynamic measure. All other things being equal, we would expect the fastest growth in EPS to indicate the best managed company and the best investment opportunity.

Return on equity is defined as the ratio of net earnings to total stockholder's equity and is a measure of the base rate of return on investment in the company by shareholders. The equity base includes proceeds from the original sale of new shares in the company (but not subsequent market transactions of existing shares) and retained earnings. ROE normally remains reasonably constant for mature companies. It is a reasonably good measure of how efficiently management is using the shareholder's resources.

Finally, market sector plays an important role in determining profitability of some companies. For example, the profitability of a whole sector may be affected by factors totally outside the control of individual management. For example, foreign competition with an input factor advantage, say lower labor costs, can reduce the profitability of even efficiently managed companies in the same sector. A less efficiently managed company in another sector may be more attractive.

Our strategy, then, is as follows. We would like to invest in the company with the highest EPS growth, assuming we can achieve a better than risk-free return. However, we are uncomfortable in investing on the basis of EPS growth alone, because management efficiency and/or sector economics may affect this factor. Since both ROE and market sectors are more stable than EPS growth itself, it would seem to make sense to see if either or both variables accounted for the observed EPS trends. If, for example, the sector accounted for much of the EPS variations, we would look for the best performing company within the best sector.

Specifically, we will look at the five-year average percent change in EPS for eight companies in each of four sectors. The eight companies have the highest average ROE over the past five years. These data will be set out (below) in a table with rows being market sector and columns being the ranking of average ROE within the sector (1 is first, etc.).

Our purpose in this analysis is to determine if either market sector or ROE explains the observed variations in EPS. If neither of these two factors influences the EPS figures, then we might as well base our investment decision on the EPS figures alone. However, if one factor influences the EPS values more than the other, then we should weight that factor more heavily in our decision than the other. This sort of analysis is termed an "analysis of variance" or ANOVA for short. Once this analysis is completed, we will know whether it is best to choose a company on the basis of either sector and EPS, ROE and EPS, or EPS alone.

Our table is listed below. These values are the percent changes in average EPS of several companies from 1980 - 1985. Again, the companies were chosen because they had the highest five-year average ROE, arranged by column with column 1 being the highest. The rows span market sector as follows:

- row 1: Computer companies
- row 2: Drug companies
- row 3: Consumer appliance manufacturers
- row 4: Oil companies

row#								
1:	39.2 ,	29.3 ,	16.6 ,	57.3 ,	37.2 ,	16.7 ,	15.8 ,	21.9
2:	12.1 ,	9.3 ,	23.7 ,	17.7 ,	14.3 ,	8.9 ,	5.1 ,	3.7
3:	9.1 ,	7.8 ,	9.6 ,	10.2 ,	12.7 ,	4.8 ,	1.6 ,	5.8
4:	-12.2 ,	2.9 ,	-7.0 ,	-5.5 ,	-11.8 ,	-2.6 ,	-17.8 ,	-7.0

TABLE 6.1 *Values of average EPS as a function of ROE and market sector*

6.1.2 PERFORMING THE ANALYSIS

The first step of this analysis is to put the data into a desk calculator Variable. This can be done in one of two ways:

Option A: The ASCII file STOCK.ASC contains the data in Tables 6.1, 6.4A, and 6.4B. You may use the File I/O menu to convert this data into ASYSTANT format and then read the converted subfiles into Variables. Appendix A provides step-by-step instructions on how to do this. After the data have been read into Variables, select Stats from the Main Menu [refer to Screen 6.1].

Option B: You can type in the data manually. Select **Stats** from the Main Menu. This will place you in the Statistics Menu [refer to Screen 6.1]. Select the **Edit** option and enter **Z** for the prompted Variable. Enter 4 for the number of rows, 8 for the number of columns. You will be placed in an editor, exactly like the array editor that has been introduced in earlier sessions. Enter the data in Table 6.1. After you enter the last number, press **Enter** a second time to return to the Statistics menu.

Tabular Statistics		
Menu Options		Variables
Edit	Basic Stats	R= .0000
Sort	Percentiles	S= .0000
Hyp Tests	Histogram	T= .0000
Distrib's	ANOVA	U= .0000
Correlation	Regression	V= .0000
Return		W= .0000
		X= 8 x 5 REAL
		Y= 8 x 1 REAL
		Z= 4 x 8 REAL

SCREEN 6.1

6.1.3 HYPOTHESIS TESTING

Before we proceed with the analysis of variance, we must verify that the data is normally distributed, since this is a requirement for proper interpretation of the probabilistic statements made by the analysis. This hypothesis is tested by selecting the **Hyp Tests** item.

Tabular Statistics	
Hypothesis Testing	Variables
KS Normality test 1 Sample t test 2 Sample t test 1 Sample Chi-sq test 2 Sample F test Wilcoxon Signed-rank test Mann-Whitney Rank-sum test Return	R= .0000 S= .0000 T= .0000 U= .0000 V= .0000 W= .0000 X= 8 x 5 REAL Y= 8 x 1 REAL Z= 4 x 8 REAL

SCREEN 6.2

You will be offered a menu of hypothesis tests. [refer to Screen 6.2]. We are interested in the KS Normality test at the top of the menu. The "KS" in this item is an abbreviation for the inventors of this type of test, Kolmogorov-Smirnov, and it is a nonparametric test for "goodness-of-fit", i.e., how well a set of data fits a hypothesized distribution.

Upon selection of this menu item, you will be placed in a prompt list allowing you to choose a subset of a Variable array on which to perform the test. We wish to test the entire table in the Z Variable, so select Z for the Variable and accept the default row and column values, which span the entire Variable, by pressing Esc.

Press Esc again to continue. The prompt list that appears allows you to tailor this test by selecting a test significance level and an alternative hypothesis [refer to Screen 6.3]. The significance level is a measure of the probability of rejecting the null hypothesis (in this test, that the data is normal) when in fact it is acceptable (i.e., the data is normal). This rejection is known as a "Type 1 error". Thus, the larger the significance level, the more often a Type 1 error will occur. Usually, levels of 0.10 through 0.005 are used. For our example, we will accept the default 0.01 significance level.

Kolmogorov-Smirnov Test		Test specifications
Select variable (R-Z): $\langle Z \rangle$		Significance level: $\langle .0100 \rangle$
Start	End	Less/Not equal/Greater: $\langle N \rangle$
row: $\langle 1 \rangle$	$\langle 4 \rangle$	Quit/Continue: $\ast Q \ast$
col: $\langle 1 \rangle$	$\langle 8 \rangle$	
		H0: Distribution is normal
		H1: Distribution is not normal
Continue/Quit/New Parms: $\ast C \ast$		
Kolmogorov-Smirnov Results		
Var 1 stats		
Mean: 10.2937		
St dev: 15.7554		
# points: 32		
Test stat D: .1316		Accept H0: Distribution is normal
P(Z > D): > 0.2		at significance level: .0100

SCREEN 6.3

The alternative hypothesis is the hypothesis against which the null hypothesis is tested. You are probably familiar with alternative hypotheses in t tests used in the comparison of two mean values. There, the null hypothesis is that the two means are equal and typical alternative hypotheses are that the first mean is greater than, less than, or not equal to the second mean. In this normality test, the alternative hypotheses are indicative of the shape of the distribution and the choices are skewed left, skewed right, or simply not normal. We are only interested in this latter possibility now; if the data are not normal, we will not be able to interpret the ANOVA results properly. Thus, accept the default alternative hypothesis of Not equal, and continue with the test by selecting Continue. The results that will appear on the screen are shown below:

```

Var 1 stats

Mean:      10.2937
St dev:    15.7554
# points:  32

Test stat D: .1316      | Accept H0: Distribution is normal
P(Z > D):    > 0.2      | at significance level: .0100
  
```

TABLE 6.2 *Kolmogorov-Smirnov normality test results*

These results indicate that the data are normal, so we can proceed with the analysis of variance. Now return to the Stats Main menu (Esc, Esc, and select Return).

6.1.3 ANALYSIS OF VARIANCE

As previously described, the ANOVA will indicate whether it is best to choose a company on the basis of either sector and EPS, ROE and EPS, or EPS alone. For the two-factor analysis of variance we are interested in, the analysis is based on a model of the form:

$$Y(ij) = \mu + a(i) + b(j) + \text{res}(ij)$$

where the $Y(ij)$ are the actual data values and μ is the mean of the Y 's. The $a(i)$ represent the mean deviations of the Y 's rows, and the $b(j)$ represent the mean deviations from along columns, with the $\text{res}(ij)$ being the leftover residuals. These deviations are used to partition the total sum of squares found in the Y 's into that contributed by the deviations in the $a(i)$, $b(j)$, and $\text{res}(ij)$. If the factors along rows and columns did not affect the Y values, then it would be expected that the sums of squares for these terms should be approximately the variance of the Y 's weighted by the number of rows, columns, or rows times columns (cells). However, if one factor explained nearly all the variations in the Y 's, then that factor would have a disproportionate amount of the total sum of squares. Statistical statements about the size of the partitioned sum of squares can be made by dividing these sums by the residual sum of squares. These ratios are F distributed with appropriate degrees of freedom. If a factor explains a large portion of the variations in the Y 's, the F -value will be large, indicating that the factor significantly affects the Y values. The term "significantly" can be quantified by computing the probability of obtaining a larger F -value. This probability is termed the "significance"; a small significance value for a given factor implies a large likelihood that the factor does influence the data values.

We can determine if market sector or rank in ROE has any effect on the EPS growth figures by subjecting our data to an ANOVA and looking for large F -values (small significances). Select ANOVA. You will be placed in another menu of ANOVA options. These options are explained in Reference Chapter 11. Our problem requires selecting the Table ANOVA option. You will be requested for a Variable in which the table resides. Enter the Variable Z here, if it is not already defaulted, and select Continue. You may write out the results to a printer, disk (note file), or the screen. Select S for screen and continue (Y). You will see the following results appear:

	Sum Sq's	Df	Mean Sum Sq's	F-Value	Significance
Fact 1	5516.5640	3	1838.8547	29.7314	.0000
Fact 2	879.8088	7	125.6870	2.0322	.0987
Resid	1298.8262	21	61.8489	.0000	.0000

TABLE 6.3 ANOVA results

The first row contains the results for factor 1, the market sector. The second row holds results for factor 2, the ranks of ROE. The results above indicate that factor 1 (market sector) has a very significant effect on the variations in the dependent (EPS) data, while factor 2 (ROE) has only a weak effect on the the variations in the dependent (EPS) data.

6.2 Basic Statistics

Since the market sector influences the EPS figures so strongly, we need to locate the sector with the highest average EPS figures. We will then select the company with the highest EPS growth within that sector for further analysis. **Quit, Return** gets us back to the Statistics menu.

Select **Basic Stats** to calculate the mean of each sector. You will be placed in a prompt list allowing you to select the Variable of interest. Select **Z**. Since market sectors span rows, you must set the start- and end-row values to limit the computation of the basic statistics to a single row. For example, to analyze only row 2, press the **down arrow** key and choose start- and end-row limits of 2.

Press **Esc** to continue, leaving the column start and end values spanning all columns, 1-8. Select **Continue** and once again you are presented with the option of printing the table to the screen, printer, or note file. As before, select the screen (or just press **Esc** to accept the default values). The results are displayed as in Screen 6.4.

Basic Statistics			
Select variable (R-Z): (Z)		Maximum:	23.7000
Start	End	Minimum:	3.7000
row: (2)	(2)	Sum:	94.8000
col: (1)	(8)	Mean:	11.8500
		Median:	10.7000
		Sample Variance:	43.9857
		Standard Deviation:	6.6322
		Skewness:	.6450
		Kurtosis:	-.0211
		Sum of Squares:	1431.2802
		Root Mean Square:	13.3757
Continue/Quit/New Parms: * Q *			
Menu Options		Variables	
Edit	Basic Stats	R= .0000	
Sort	Percentiles	S= .0000	
Hyp Tests	Histogram	T= .0000	
Distrib's	ANOVA	U= .0000	
Correlation	Regression	V= .0000	
Return		W= .0000	
		X= 8 x 5 REAL	
		Y= 8 x 1 REAL	
		Z= 4 x 8 REAL	

SCREEN 6.4

Notice that the "Mean" change of EPS values is 11.850. In this example, the mean is equivalent to the average percent. Select New Pirms to analyze each row, 1-4, in turn. Then Quit to the Stats Main menu. You should find average percent change EPS values of:

row #	avg % change in EPS
1	29.250
2	11.850
3	7.700
4	-7.625

Apparently, the first group, computer companies, has the greatest average EPS growth rates. You might also run a two- sample t test on the first and second groups to verify that the mean of group 1 is significantly different than the mean of group 2. This hypothesis test can be done with the "2 sample t test" item in the "Hyp Tests" menu.

We have completed the first portion of this Session. We have found that the computer industry is probably the most profitable sector in which to invest. In the next section, we will analyze the growth of a particular computer company with the goal of predicting its share price in 1986. If our expected return exceeds 8%, we will purchase it.

6.3 Prediction by Regression

6.3.1 BACKGROUND

In this section we will try to predict the stock price of a company a year in the future, using a set of historical statistics for that company of the past eight years. The main tool used for this analysis will be the "Regression" option in the Statistics menu.

The idea of regression analysis is to build up a linear model that explains the relationship between a single dependent variable and a series of independent variables. In this example the dependent variable is the average stock value in year $n+1$, and the independent variables are the vital corporate statistics measured in year n . The past history of the company gives us a set of experimental results which can be used to establish values for the model parameters. Once the parameters have been evaluated the model can then be used to predict future stock prices.

Please Note: This is only an illustrative example of the use of regression analysis. The data is hypothetical and the method is not a realistic technique for predicting future stock prices.

The data that will be used for this example are listed below. The independent variables are:

Year	Avg Share Price (\$)	Total Sales (million \$)	Net Profits (million \$)	Earnings/Share (\$)	St&Poor Avg (\$)
1977	.6300	.5000	-.1200	-.0100	119.3000
1978	.5300	1.1000	-.5000	-.0600	138.6000
1979	.9000	2.2000	.1200	.0100	121.7000
1980	1.0300	6.3000	.2300	.0200	141.3000
1981	1.9700	8.3000	.7280	.1000	142.7000
1982	2.6300	15.4000	.9650	.1300	131.3000
1983	4.8600	29.8000	2.9000	.3300	121.5000
1984	13.2000	39.2000	4.6100	.5000	159.5000

TABLE 6.4A *Historical Statistics for Computer Company*

and the dependent variable is next year's average share value:

Year	Avg Share Price (\$)
1978	.5300
1979	.9000
1980	1.0300
1981	1.9700
1982	2.6300
1983	4.8600
1984	13.2000
1985	17.2000

TABLE 6.4B *Dependent Variable (Year N+1 Share Price)*

6.3.2 PERFORMING THE ANALYSIS

The first step will be to enter these data into two of the (R-Z) calculator Variables.

Option A: If you followed the directions of Appendix A, you should already have the matrix of historical values in Variable X and share prices in Y.

Option B: Select Edit in the Stats menu and enter the numbers manually, with the historical values in an 8 by 5 array in X, and the share prices in an 8 element vector Y (don't include the Years in either array).

Select the **Regression** option in the Statistics menu. Prompts will appear in the upper left hand corner of the screen requesting you to specify the (R-Z) Variable that contains the independent terms for the regression analysis. Each of the columns of this Variable will contain a different term, up to a maximum of 20. If the Variable array contains more than 20 columns then only a subset of the columns can be included in the analysis.

Select X for the independent terms and then press Esc to accept all columns for analysis.

Additional prompts will now appear requesting specification of the dependent variable. You may choose either a one-dimensional array or a column from a two-dimensional array. In either case, the dependent variable must be of the same length as the columns of the independent terms.

Select Y for the dependent term, and once again press Esc. Now Continue. The screen will be repainted with status and diagnostic information that is updated with each regression step. [refer to Screen 6.5]. At the beginning of the procedure the model is empty, and it will be up to you to add or delete terms from the model according to the information that is reported. The prompts at the bottom of the display allow you to enter or remove terms from the model, as well as terminate the regression procedure.

In Model Vars — Value — St Dev — F-to-rem — Sig					Overall Model	
					Deg's freedom	0
					Residual SS	2.828E2
					Regression SS	0.000E-1
Not In Vars	Part corr	Tol	F-to-ent	Sig		
1	.9276	1.0000	37.0078	.0009		
2	.9919	1.0000	3.652E2	1.327E-6	y intercept	5.2900
3	.9878	1.0000	2.410E2	4.520E-6		
4	.9850	1.0000	1.960E2	8.277E-6		
5	.4253	1.0000	1.3253	.2935	r^2 corr coeff	0.000E-1
					F-all value	0.000E-1
					Significance	1.0000

Enter/Remove: < E >
Variable x < 2 >

Quit/Continue: * C *

SCREEN 6.5

To enter a term into the model, type E in response to the prompt

Enter/Remove: < E > _

The next prompt, "Variable x", is used to select a term to fit into the model. The term numbers correspond to the column index of the independent Variable. The ASYSTANT automatically chooses as the default the term that will contribute the most to the fit of the model. This is the term that has the highest "F- to-ent" (i.e., F to enter) value or, in other words, the lowest "Sig" (significance). In this case term 2, the Total Sales in year N-1, is the default. You can override this selection by entering a different term, or you can accept it and append the term to the model by pressing "Esc".

Press the Esc key and notice how the display is updated. Term 2 has been moved from the "Not in Model" list to the "In Model" list, and all the diagnostic information has been updated to reflect this change. The

information displayed on the right side of the screen gives an overall assessment of the current model, which at the moment only includes term 2. Notice that the value of the Regression SS (sum of squares) is 278.2063 which is very large in comparison with the Residual SS (4.5701). The Regression SS is the component of variations in the dependent Variable that is explained by the model, and the residual sum of squares is the component that remains unexplained. The multiple r^2 correlation coefficient is equal to the ratio of the Regression SS to the total SS, i.e.,

$$r^2 = \frac{\text{Regression SS}}{\text{Regression SS} + \text{Residual SS}} = \frac{278.2063}{278.2063 + 4.5701} = .9838$$

This coefficient represents the proportion of variations that are explained by the model. Since r^2 is very close to 1, the single term contained in the model (total sales) is very highly correlated with the future share value. As a single statistic it provides the best indicator of future stock value.

The F-value for the overall model is equal to the ratio of the Regression SS to the Residual SS, where both the numerator and denominator are weighted by their respective degrees of freedom. More specifically,

$$F = \frac{\text{Regression SS/df}}{\text{Residual SS/(n-df-1)}} \quad \begin{array}{l} \text{where } n = \text{the number of data points} \\ \text{df} = \text{the model degree of freedom} \end{array}$$

The value of F for the overall model is listed under the heading "F-all value".

The F-value is used to determine if the proportion of variations that are due to regression is significant, given the number of degrees of freedom in the model. The "Significance" of F is equal to the probability of the F-statistic assuming a value that is greater than the measured F-value. The smaller the significance the more readily you can reject the hypothesis that the model parameters are all 0. In this case the reported "Significance" value is exceedingly small.

The actual model parameters are given by the first column of the "In model" table, together with the "y-intercept" that is listed on the right. Thus, the one term model based on total sales is given by the formula

$$\text{Next yr's avg share value} = -.3275 + .4372 * (\text{this yr's tot sales})$$

Two other columns of the "Not in" table are the partial correlation ("Part Corr") and tolerance ("Tol") values. The partial correlation is the correlation of the term to be entered with the residual vector for the current model, and the tolerance is a measure of the correlation between the term to be entered and the other terms already in the model. (The tolerance is actually equal to one

minus the square of this correlation.) It is desirable to enter terms that are highly correlated with the residuals, but not highly correlated with the terms already in the model. A useful formula that relates partial correlation with r^2 is:

$$r'^2 = r^2 + (1-r^2)R_i^2$$

where r^2 = The Multiple Correlation Coefficient
before entering the i-th Variable

r'^2 = The Multiple Correlation Coefficient
after entering the i-th Variable

R_i = The Partial Correlation of term i.

In this example, the one-term model is probably the best. If you wanted to add a second term, the term with the best (highest) F-to-enter value is Term 3. However, you can see that the "Tol" for Term 3 is quite low, suggesting Term 3 is strongly correlated with the Term already in the model. In addition, "Sig", while the lowest of the "Not in Model" terms, is still quite high. There is roughly a 29% chance that a random vector would improve the model as much as including Term 3.

However, if you still wanted to enter the term in the model, simply press the "Esc" key. This will automatically enter the term that has the most significant "F-to-enter" value (i.e., smallest "Sig"), which in this case is term 3. A similar "F-to-remove" (F-to-remove) statistic is reported along with its corresponding significance for each term in the "In Model" table, and the default selection for removing a term is based on this parameter.

At this point you might find it useful to just play around with the model, entering and removing terms and observing the effect that this has on the various fit parameters. Note that as you continue to add Terms, the overall significance value of the model gets worse even though the overall r^2 increases. In this case a one-term, or at most, a two-term model can be justified.

Note that last year's stock price is actually the worst predictor of the next year's. This surprising result is explained by the large year-to-year swings in the price.

TIP

It is generally a good practice to start the regression analysis with all the parameters included in the model, and then to remove terms to get the most significant and parsimonious fit.

Now let us make a prediction based on a two parameter model that includes term 2 (total sales) and term 3 (net profits). Enter and Remove terms from the "In Model" table as needed so

that it is left with just terms 2 and 3. Taking the coefficients from the "In Model" table, and taking the "y- intercept" from the column of "Overall Model" parameters, the model is:

$$\text{Next yr's avg share value} = .2122 + .2841 * (\text{this yr's tot sales}) \\ + 1.278 * (\text{this yr's net profits})$$

Supplying the statistics:

Year	Avg Share Price (\$)	Total Sales (million \$)	Net Profits (million \$)	Earnings/ Share (\$)	St&Poor Avg (\$)
1985	17.20	70.7	10.200	1.04	164.2

then lets us use the model to predict next year's average share value. The result is:

$$.2122 + (.2841 * 70.7) + (1.278 * 10.2) = \$33.33$$

To exit from the regression analysis enter a response of **Q** for the Quit/Continue prompt. You will then be offered the option of saving the final results of the regression in calculator Variables. Since we have no reason to save these results, press the **Esc** key twice and you will be returned to the Statistics menu. From there you may select **Return** and find yourself back in the Desk Calculator Main Menu.

THE HISTORY OF THE

REIGN OF

CHARLES THE FIRST

IN THE

SEVENTEENTH CENTURY

BY

JOHN RUSSELL

OF LINCOLN'S INN

ESQ.

LONDON

Printed by J. Sturges, at the

Printers, in Pall-mall

1719

Price 1s. 6d.

per Volume

in 4 Volumes

bound in

Red Leather

with

Gold

Stitching

SESSION 7: **DIFFRACTION PATTERNS AND TWO- DIMENSIONAL PLOTS**

This session will introduce additional array operations and plotting capabilities. We will be generating axonometric and contour plots of the diffraction pattern produced by a circular aperture. The following features will be used:

Desk Calculator

*to create, initialize, and search arrays,
to extract subarrays, and
to compute a Bessel function on an array;*

Graphics

to map and manipulate axonometric and contour plots.

7.1 Generating a 2D Array to Represent the Pattern

A diffraction pattern is an image produced by projecting monochromatic light through a small aperture. If the aperture is circular, and far enough away from the imaging screen, the image intensity will vary across the screen according to the expression:

$$|J_1(r)/r|^2$$

EQUATION 7.1

where J_1 is the Bessel function of order 1, and r is the radial distance from the center of the screen.

If you are not familiar with Bessel functions, don't worry about the mathematics for now. Bessel functions are useful in certain kinds of analysis, just as trigonometric functions are in many others. They are analogous to -- though different from -- a sine function.

To plot this pattern we will use a two dimensional array to represent the screen. Each pair of array indices will prescribe a position, and the value of the corresponding array element will represent the intensity of the diffraction pattern at that position on the screen.

Lets get started! Boot up the ASYSTANT and enter the Desk Calculator.

We begin by creating the two dimensional array described above. We will do this in two steps. First, we will create an array of r -values. The ij -th element of this array will be set equal to the distance from the ij -th position to the center of the screen. Second, we will apply $|J_1(r)/r|^2$ to each array element.

Type the following to create a 20 by 20 element array and store it in the R Variable:

```
Command: 20 20 nm:ramp store R
```

We don't actually want to use the index values that were assigned by `nm:ramp`, we only used this command to set up the array.

The r -values to be entered in this array will be generated through the formula

$$r = \sqrt{x^2 + y^2},$$

where $x=0$ is the absolute center of the screen, $x=-10$ is the left edge, and $x=+10$ is the right edge. In physical terms, you can think of our array as representing a 20 by 20 grid superimposed on the screen, with a sample taken at the center of each block. This midpoint of any box on the left edge would be $x=-9.5$, and on the right edge $+9.5$. To represent the x coordinates of each element in the array,

we should fill all the rows of the R-array with the values (-9.5,-8.5,...,-.5,.5,...,8.5,9.5). These will be the x-coordinates assigned to the j-indices. The value of 0 was intentionally omitted because the expression $|J_1(r)/r|^2$ is difficult to compute at $r=0$. To enter these values, type:

Command: 20 n:ramp 10.5 - R =

Notice we used the "=" operation, *not* "store", at the end of this command line. "=" took the lower stack entry (a 20 element vector) and assigned it to the 2-D array in R. To do this it automatically copied the vector into each row of R. "store" would have simply erased the old contents of R and replaced it with the new vector. In general, the "=" operation can be used to assign numbers to Parameters, numbers to Variable arrays, or arrays to Variable arrays. If the source stack entry is of lower dimension than the target array, the source will be copied repeatedly to fill the target. If the target is incompatible with the source (eg., trying to assign an array into a scalar), "=" will generate an error.

Now let's square the contents of the R-array to get the x^2 values:

Command: R R * R =

Here again we have used the "=" operation to replace the contents of R with the squared x-values, although here "store" and "=" are equivalent because the source and target have the same dimensions and data types.

Notice that the y^2 values are exactly the same as the x^2 , except that $x(i,j)=y(j,i)$. Thus, to produce an array of y^2 -values we need only transpose the rows and columns of the array of x^2 -values. This can be done using the "trans" command. Thus, to set up the array of r-values type:

Command: R R trans + sqrt R =

This completes the first step. To produce the final pattern we must now apply Equation 7.1 to the elements of R. Computing Bessel functions is a built-in feature of the ASYSTANT. The "Jn" operation appears as an option in the "Conversions and Special Functions" menu, but it can also be executed in a command line. The lower stack entry specifies the order of the Bessel function, and the top stack entry provides the function argument. Thus, to compute $J_1(r)$ for each element of the R-array, type:

Command: 1 R JN

There will be a pause while this calculation is performed. To compute the Bessel function using the menu, you would place "1" then "R" on the stack and select the "Jn" option.

Now to finish the computation we must divide this result by R, and then square it. We use the option "dup", which duplicates the top item on the stack. Thus the sequence "dup *" squares any number on the stack. The final result will be saved in Z.

Command: R / DUP * store Z

While we're at it, let's extract a 10 by 10 subsection from the center of the array. We'll use this in the graphics menu to plot a 2x expanded view. Type Z Enter to put an image of Z on the stack. Now, using the calculator menu (not the command line) select sub on the Array menu [refer to Screen 7.1].

Stack Contents		Parameters	
Z	DP-REAL DIM[20 , 20]	A=	.0000
		B=	.0000
		C=	.0000
		D=	.0000
		E=	.0000
		F=	.0000
		G=	.0000
		H=	.0000
		I=	.0000
Extract Subarray		Variables	
Row	Start: (6)	R=	20 x 20 REAL
Dimensions	Length: (10)	S=	.0000
	Incr: (1)	T=	.0000
Column	Start: (6)	U=	.0000
Dimensions	Length: (10)	V=	.0000
	Incr: (1)	W=	.0000
		X=	.0000
		Y=	.0000
		Z=	20 x 20 DP-REAL
Press Esc when done			

SCREEN 7.1

Please note: "sub" is available from the command line, but requires several parameters. When executed from a menu, a prompt list is provided, making it easier to use. If your work causes you to use "sub" frequently, you may want to memorize the command syntax. Refer to Reference section 2.7 for details.

The central 10x10 cross section starts at (6,6). Under "Row Dimensions", enter 6 for the Start, and 10 for the Length and leave the Incr prompt at 1. Enter the same values for the corresponding "Column Dimensions". Press Esc when all entries are correct.

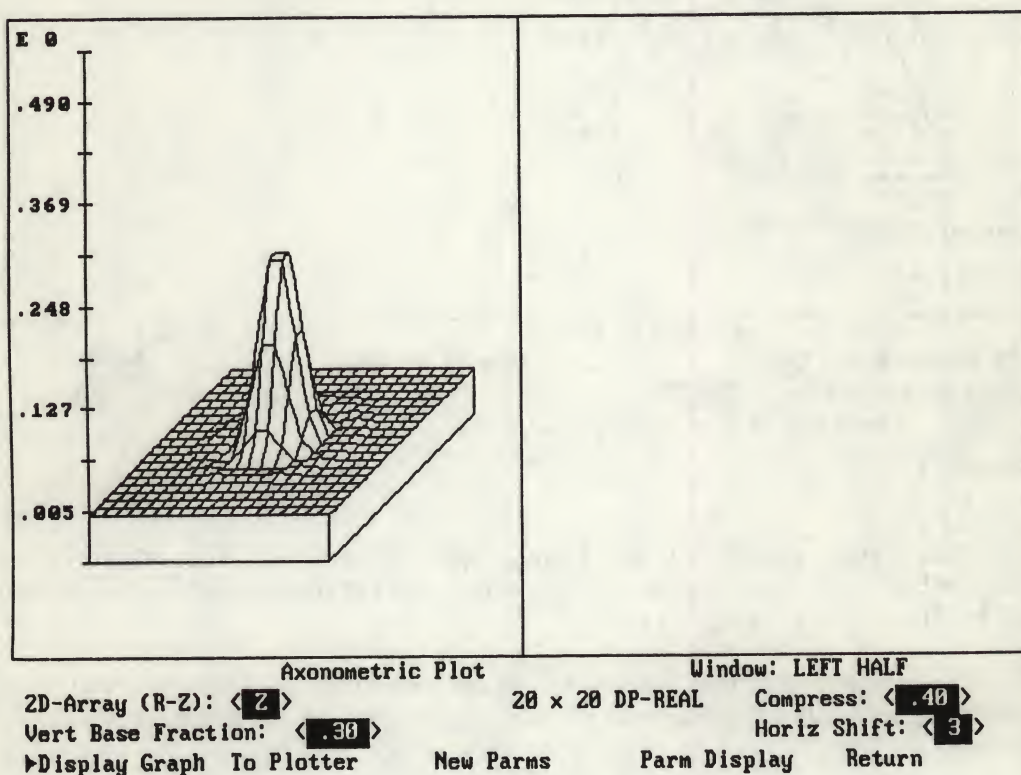
Now use the command line to store the 10 by 10 subsection in the Variable "Y".

Command: STORE Y

7.2 Axonometric and Contour Plots

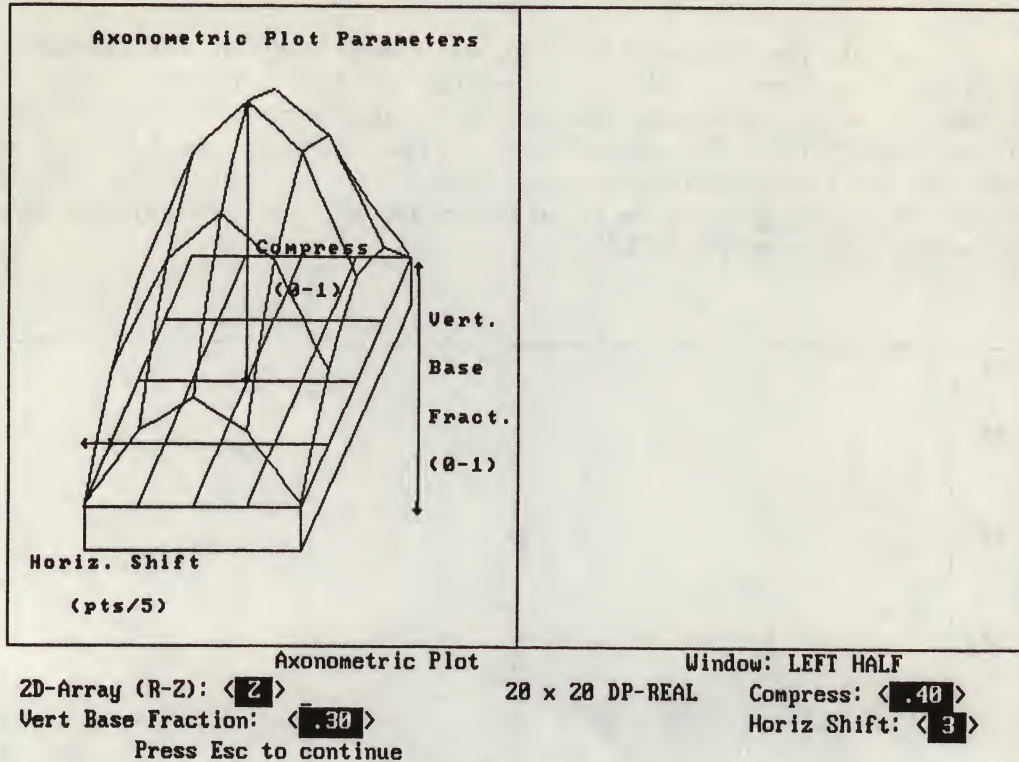
Let's now use the graphics menu to generate axonometric plots of the Z and Y arrays. Press **PgDn** to go to the main menu, and execute the **Graphics** option. From the main Graphics Menu, select **Window**, and change the plotting window to the **Left Half**. Then from the Graphics Menu select **Axon**.

"Axon" asks you to select a Variable, which must be a two dimensional array. Selecting a vector will generate an error message. Choose **Z**. "Axon" prompts for several parameters that control the appearance of the plot. These are explained fully in Reference Chapter 4. For now accept the default values. Press the **Esc** key to continue on to the menu of "Axon" options. The "Display Graph" option will generate the actual plot on the display. Select **Display Graph** and watch! [Refer to Screen 7.2]



SCREEN 7.2

Execute **Parm Display**. This will produce a demonstration plot showing the significance of each display parameter. It will help you to understand how changes in these parameters will effect subsequent plots. If you want to try any changes, execute **New Parm**s [refer to Screen 7.3].



SCREEN 7.3

Now **Return** to the Graphics Menu. Select **Window** and **Top Right**. We will use the top right hand quadrant to generate an axonometric plot of the central 10 by 10 subarray we stored in Y.

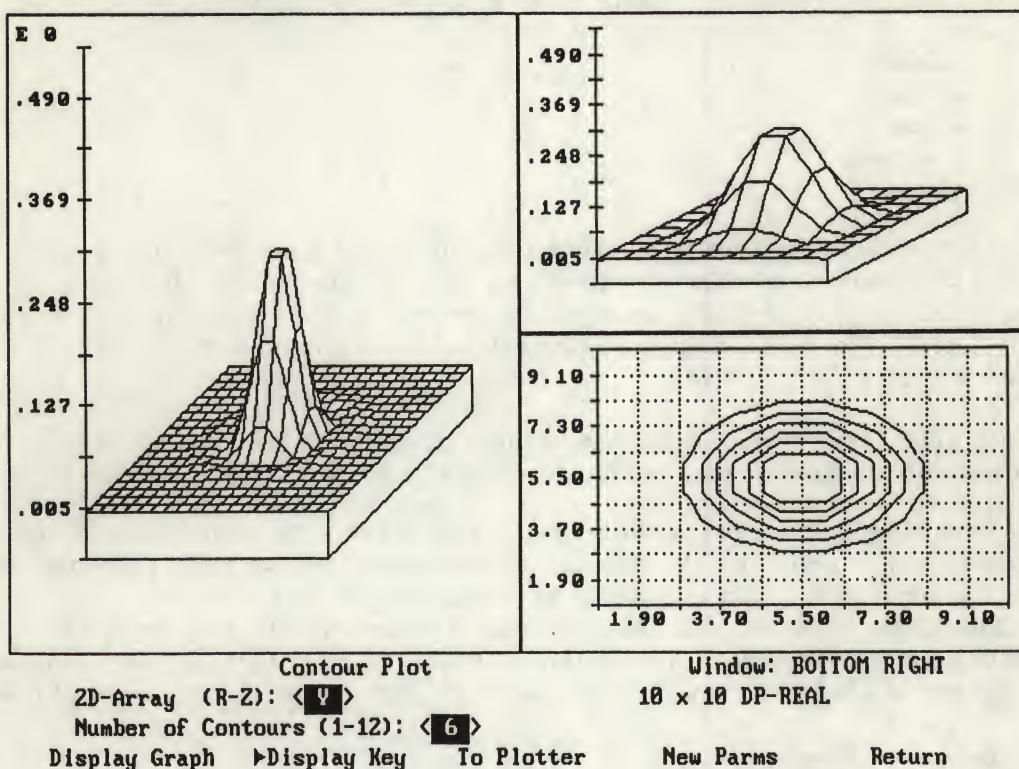
Use **Axon** to plot the Y-array in the current window. This plot will be a twofold expansion of the central portion of the plot on the left half of the screen.

There is another way of plotting two dimensional data called "contour plotting". This produces a "relief map" of the array, where the relief lines trace out paths of constant amplitude. This type of plot can be generated using the "Contour" option. But before selecting this option, first change the current plotting window to the **Bottom Right** quadrant and, if you are using an EGA card, go into the **Colors** option to change the "Vuport Color" to 0 if it is not already. (Make sure you change the window before changing color).

Note: Color is particularly useful in interpreting contour plots. If you are not using an EGA card or compatible, contour plots will be difficult to interpret on the screen. Try generating a color plot on a digital plotter.

Now execute **Contour**. You will be asked to select the data array, and to choose the number of contours to be displayed. The contour values will scale automatically to be equally spaced between the maximum and minimum data values. For example, if 4 contours were selected and if the data ranged between 0 and 1, then the contours would be assigned the values 0.2, 0.4, 0.6, and 0.8. For this example, try plotting 6 contours using the Y- array.

Once you have completed the prompt list, press the **Esc** key to move on to the "Contour" menu. To generate the plot execute the **Display** option (or **To Plotter**). You can generate a listing of the contour values by executing **Display Key** [refer to Screen 7.4].



SCREEN 7.4

It is often useful to show contour and axonometric plots simultaneously. This allows you to interpret 2D-data from two different perspectives.

7.3 Integrating the Intensity over the Peak

Before leaving this example, we will exercise two more array functions: "n:search" and "lookup". Suppose we want to integrate (or sum up) the intensities in the Z-array, but only for those elements that are greater than 0.05. These elements cannot be separated out by taking a subarray because they don't constitute a rectangular portion of Z.

To proceed with this calculation, return to the Desk Calculator and use the next option to find the menu of "Array Operations." The "n:search" operation is used to search an array for those elements that satisfy a special condition.

To use "n:search", first put the Z array on the stack, followed by the number .05. Now execute n:search. The following submenu will be displayed:

```
=n:search
<n:search
>n:search
<>n:search
return
```

Move the cursor down to the >n:search option and press the Enter key. The Z array will be searched for all entries that are greater than .05, and a list of the indices of these n elements will be reported on the top of the stack as an n by 2 array. In this example, 16 elements of Z are greater than .05, so the result will be a 16 by 2 array.

Please note: The "n:search" command cannot be executed in a command line, but each of the specific searches (eg., "=n:search", ">n:search", etc.) can be.

This 16 by 2 index list specifies the locations of the elements of the Z array that we want to pull out. The "lookup" operation will extract these elements. It uses the array on the top of stack as an index list, and pulls out the corresponding elements from the array that is underneath it. The result will always be one dimensional. To use this operation we must first put the Z array on the top of stack and "swap" it with the index list. Type in the command line:

Command: Z swap lookup

Now all that remains is to sum up the elements of the array on the top of stack. This is accomplished as follows:

Command: []sum

"[]sum" computes the cumulative sum of the input array. The result 2.1983 should now be displayed on the top of stack!

SESSION 8: **CONFIGURING YOUR HARDWARE FOR DATA ACQUISITION**

In order for ASYSTANT+ to operate properly you must configure and install your data acquisition board according to the manufacturer's specifications and then configure ASYSTANT+ accordingly. We have set aside this session to help you do this.

In this session we will introduce the **Configure** Option in the **Acquire** menu to set up ASYSTANT+ to operate with your hardware.

Needed for this example session

- Compatible A/D hardware
- Manufacturer's Manuals for A/D hardware
- Appendices B & C of this manual
- Small tools (if it becomes necessary to open your PC)

Note: The **Acquire** option is only available in ASYSTANT+; users of ASYSTANT are not able to enter the **Acquire** menu of data acquisition routines.

8.1 Getting started

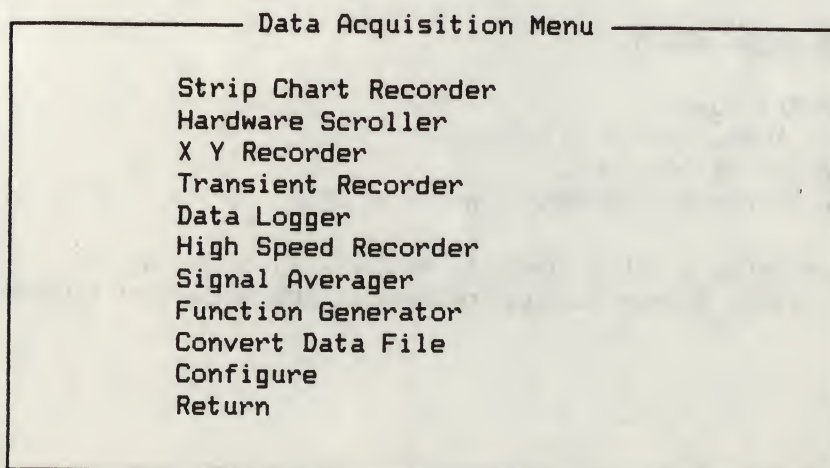
ASYSTANT+ assumes that your board (or system) will have between 4 and 16 A/D input channels, 2 D/A channels, 8 bits of digital input, and 8 bits of digital output. (There are some exceptions; see Appendices C & D for details specific to your hardware).

Most data acquisition boards allow the user to select the voltage range to be used for A/D input and D/A output. You may also have a choice of hardware gain. Depending upon the A/D converter used you may have a choice between 2's complement and offset binary integer numbers (i.e., -2048/2047 vs. 0/4095 for a 12-bit A/D). Your choice for these options, and others, will be used to initialize ASYSTANT+.

If you have not already done so, set all DIP switches and/or jumpers on your data acquisition board according to your needs and hardware specifications (consult Appendix C and your board manufacturer's instructions if you are unfamiliar with this process).

Now you must configure ASYSTANT+ for your data acquisition system. When you load ASYSTANT+, the configuration file ACQUIRE@.CFG found on your System Diskette is read and ASYSTANT+ is initialized accordingly. *You must setup this file for your particular system before you acquire any data with ASYSTANT+.* Once ACQUIRE@.CFG is properly initialized, it simply must be available to ASYSTANT+ at load time, i.e., on your hard disk or on your System Diskette. You will not need to alter it again unless you change your acquisition system.

So, load ASYSTANT+ (the default values stored in ACQUIRE@.CFG will be used for now). Then select the **Acquire** option in the Main Menu to display the Data Acquisition menu [refer to Screen 8.1].



SCREEN 8.1

If you are using a floppy based system, insert the Acquire 2 disk (violet label) in the System overlay drive (drive A). Acquire 2 contains the overlay files for the bottom five options of the menu; Acquire 1 overlays are associated with the top five options. You will need to insert the appropriate diskette as you move from option to option. In this case, we will select "Configure" which is the last option (other than "Return") so we need the Acquire 2 diskette.

Choose the **Configure** option. Now, since we are going to modify the ACQUIRE@.CFG file, it must be present in the System Overlay drive. On a floppy system, remove the Acquire 1 Diskette and insert the System Disk in its stead. Choose **Data Acquisition Board** from the "Acquisition Configuration" Menu. You can use either the arrow keys and "Enter" as in other ASYSTANT menus, or you may use the function keys ("F1" in this case). The current (in this case, default) configuration for data acquisition will be displayed [refer to Screen 8.2].

```
----- Data Acquisition Hardware Configuration -----  
  
Current configuration: Data Translation DT2800  
  
Base address (hex): 2EC  
  
16 A/D channels  
    0 4095. Min/Max A/D values  
   -10.00 10.00 Min/Max A/D voltage input  
  
Hardware gain: 1.00E0  
  
2 D/A channels  
    0 4095. Min/Max D/A values  
   -10.00 10.00 Min/Max D/A #1 output voltages  
   -10.00 10.00 Min/Max D/A #2 output voltages  
  
Continue configuration/Quit: * C *
```

SCREEN 8.2

Press **C** to continue if you wish to change any of the configuration parameters. The first screen will prompt you for the clock speed and processor in your computer and your data acquisition board. The IBM models and the most popular compatibles are referred to by name; if yours is not listed refer to

your system documentation. [Refer to screen 8.3]. Enter the appropriate values (you can go back if you make a mistake) and press C to continue. ASYSTANT+ will now look up the default settings for the particular data acquisition hardware you specified, and display them in a second prompt list [refer to Screen 8.4]. The defaults are set to match the manufacturers' standard settings. Keep in mind that your screen will vary from the illustration depending on what board you selected.

Data Acquisition Hardware Configuration	
Computer Types	Device Types
(1) PC (8088/8086 4.7MHz clock) IBM-PC, XT, Compaq Portable, Deskpro	(1) Data Translation DT2801
(2) PC (8088/8086 8MHz clock) Compaq Deskpro, AT&T 6300	(2) Data Translation DT2801A
(3) PC (8088/8086 10MHz clock)	(3) Data Translation DT2801/5716
(4) AT (80286 6MHz clock) IBM-AT	(4) Data Translation DT2805
(5) AT (80286 8MHz clock)	(5) Data Translation DT2805/5716
(6) AT (80286 10MHz clock)	(6) Data Translation DT2808
(7) Reserved (user-defined)	(7) Data Translation DT2818
(8) Reserved (user-defined)	(8) Tecmar Lab Master
	(9) Tecmar Lab Tender
	(10) Keithley DAS 500 System
	(11) IBM-DACA
	(12) Cyborg 91-I
	(13) Metrabyte DASH16

Enter computer type: (2) Enter device type: (13)

Continue configuration/Quit: * C *

SCREEN 8.3

Configuring Metrabyte DASH-16	
Hex I/O address: (300)	Decimal equivalent: (768.)
Number of A/D channels: (16)	
Minimum A/D digital value: (0)	
Maximum A/D digital value: (4095.)	
Minimum A/D voltage input: (-5.00)	
Maximum A/D voltage input: (5.00)	
Hardware gain selected: (1.00E0)	
Number of D/A channels: (2)	
Minimum D/A digital value: (0)	
Maximum D/A digital value: (4095.)	
Minimum D/A voltage output channel 0: (.00)	
Maximum D/A voltage output channel 0: (5.00)	
Minimum D/A voltage output channel 1: (.00)	
Maximum D/A voltage output channel 1: (5.00)	
Continue configuration/Quit: * C *	

SCREEN 8.4

The biggest problem you are likely to have is determining your hardware's current configuration. Some hardware manuals are not known for their clarity, and obscure switches and jumper pins often make a dramatic difference. We've tried to summarize the salient features in Appendix C.

If you wish to change any of the default acquisition hardware settings, use the arrow keys to move the cursor next to the value to be changed and enter the new value. Parameter prompts appear in the following order:

Base address of the board (ASYSTANT+ expects the entry in hexadecimal format and displays the decimal equivalent).

Number of A/D channels (dependent on board type and whether the board is configured for single-ended or differential input).

Minimum and maximum A/D digital values (determined by the number of bits and method of representation used by the specific A/D converter).

Minimum and maximum A/D voltage input (in volts, ie., if your board is jumpered for +/- 10mV you should enter -0.01 and 0.01 respectively).

Hardware gain (should take into consideration any signal amplification that will occur before the signal reaches your board and any hardware selectable gain for which your board is configured. For example, if you were planning to use a 100X gain amplifier for signal conditioning to get your original signal from +/- 10mV to +/- 1V and a board hardware gain of 10X to get the signal to +/- 10V for your A/D input then you would enter a hardware gain value of 1000).

Minimum and maximum D/A digital values (determined by the number of bits and method of representation used by the specific D/A converter).

Number of D/A channels (only two channels will be supported, regardless of the number of D/A channels that your board may have. Each channel may be configured for a different voltage range output, if this option configuration is supported by the hardware).

When you are satisfied with every entry, press C and ASYSTANT+ will be configured to your specifications. You could have opted to quit at this point, leaving the configuration unchanged, by pressing "Q".

This completes data acquisition hardware configuration. ASYSTANT+ saves the new configuration parameters in the file ACQUIRE@.CFG. This file must be with the System Disk files when you load ASYSTANT+. The new values will be used until they are changed again by re-executing the configuration routine.

Good luck!

January 1st 1900

Dear Sir,

I have the honor to acknowledge the receipt of your letter of the 29th inst.

and in reply to inform you that the same has been forwarded to the proper authorities.

I am, Sir, very respectfully,
Yours truly,

Wm. H. Smith

Secretary

U. S. Fish Commission

Washington, D. C.

Enclosed for you are the following documents:

1. A copy of the report of the Fish Commission for the year 1899.

2. A copy of the report of the Fish Commission for the year 1900.

I am, Sir, very respectfully,
Yours truly,

Wm. H. Smith

Secretary

U. S. Fish Commission

SESSION 9: READING & PROCESSING DATA FROM A PHILIPS DIGITAL STORAGE OSCILLOSCOPE.

The PM2260 software package is especially designed to process data acquired from a Philips Digital Storage Oscilloscope (DSO). Any Philips digital storage oscilloscope can be used. Section 6 of part 1 shows an overview of currently supported oscilloscopes.

In this session a demonstration is given on how to acquire data and how to calculate a power spectrum.

To avoid misunderstandings:

With the software package you cannot operate the DSO in a remote way. It is only possible to transfer data (and stored front-panel-settings) from a DSO and eventually send the data from a PC to the DSO. Applied signals have to be adjusted using the normal functions of the digital storage oscilloscope.

The user does not need to have any specific knowledge about the GPIB interface bus.

9.1. Getting Started.

Connect the oscilloscope via the GPIB to the PC. Check the address setting of the oscilloscope (address must be between 1....30).

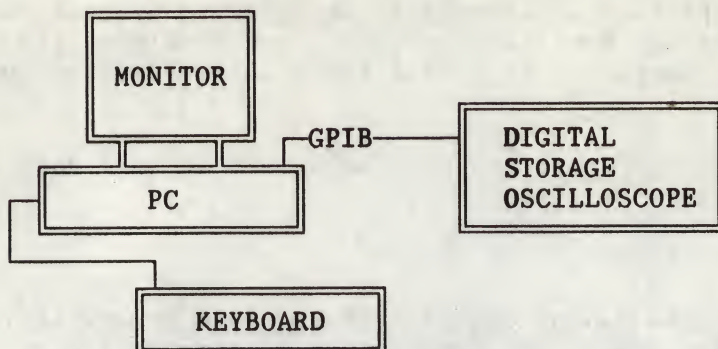


Figure : Connections of a PC + Oscilloscope.

Apply a signal to channel A of the oscilloscope. Not a sine wave, because the power spectrum of a sine wave will only show one component. Apply a triangle, sawtooth or square wave; frequency 15 kHz, amplitude 5 Vpp.

- NOTE!**
1. Another possibility is to connect the calibration output signal to channel A of the oscilloscope.
 2. For more details on how to adjust a signal on the digital oscilloscope, please refer to the corresponding operating manual. Latest models of the Philips Digital Storage Oscilloscopes are all equipped with an Autoset (green) button. Just apply a signal and press Autoset.

9.2. Acquiring data from a digital storage oscilloscope.

Start up the PM2260 software package e.g. type

OSP 3320 <Enter>

when using a PM3320 digital storage oscilloscope.

Select the following so called pull-down menus:

Instrument	Search	In order to find the address of the digital storage oscilloscope. After finding the address of instrument the next pull down menu is automatically selected.
-------------------	---------------	--

Note! This menu selection only has to be done if the program is used for the first time and the oscilloscope address has remained unchanged.
In case of doubt, select the option again.

Register	Select the register in which the data is stored (in most cases this is register 0). Move the cursor with the arrow up/down, press <Enter> to select.
Channel	Here has only a selection to be done if the DSO is in dual channel mode. In all other cases the proper channel is automatically selected.
Data_points	This selection offers the user the possibility to read the whole signal or just a part of it.
Begin 0	Start point of reading data.
End 4095	Last point to read
Count 8	Count factor: each 8 th point is read. Thus in total 512 points are read.

All selected options are displayed in the **STATUS** field. In this exercise we want to transfer 512 data-points. Therefore, **data requested** in the **STATUS** field should indicate 512 points.

For more information about **register** selection, **channel** selection or **data_points** selection please refer to the operating manual of the DSO or PART 1, section 3.2 to 3.4, of this manual.

Acquisition	From instrument	The acquisition menu allows the user to select whether data is read from an instrument or from a file.
	From file	Reading data from a file is useful when data to an instrument.

Select the item **From instrument** from the **acquisition** menu.

After selecting **Acquisition** from instrument, the message:

reading data

appears on screen.

After the data is read, the next pull-down menu is automatically selected.

Store	Unscaled	No front panel settings are stored.
	Scaled	Front panel settings (timebase, attenuator, offset) are also stored.
	Save to file	Store current data read into a file.
	Send to instrument	Data which is read from a file can be sent to an instrument.

Select **Scaled** and **Save to file** by moving the cursor to the required selections.

After selecting **Save to file** a new pull down menu appears on the screen. In this menu the user can define a filename, a directory path, ask for a directory overview, or start the saving process. In case you want to change a filename, move the cursor to **Filename**, press <Enter> and enter the filename, e.g. DEMO.ASY.

If you wish to scroll through a directory, use the <Tab> key on the keyboard to move the cursor to the pull-down menu in which the directory is displayed. With the arrow-up/down keys you can scroll through a directory. Each time up to 10 files are displayed in the windows. Return to the previous pull-down menu by using the <Tab> key again.

After defining a filename select **START SAVING** to proceed with the data-storage process. During data-storage the message **saving file(s)** appears on screen. In the **STATUS** field the name of the stored file is displayed.

After the files are successfully saved, the next pull-down menu is selected automatically. Press <Enter> to start the ASYSTANT program. During the start up of ASYSTANT enter **R** to restore pre-defined setup's and then **C** to continue with ASYSTANT.

9.3. Transfer data from a disk file to an I/O variable.

Before any calculation can be done on the measured oscilloscope data, the data has to be read from disk into the ASYSTANT environment. Select file I/O. A new menu appears on screen.

Select File/Var transfer. This option allows the user to read data from a disk file into the I/O variables of Asystant.

Press <Enter> and enter filename to restore: DEMO.ASY.

A short listing of the comments and subfile sizes are now displayed on the screen. The filename DEMO.ASY should contain only 2 subfiles of 512 points each. These 2 subfiles have to be copied from (hard) disk to an I/O variable of ASYSTANT.

Select C to continue.

The screen should now look like the picture in screen 9.1.

File Input/Output			
File Name: <C:DEMO.ASY>		Quit/Continue: * C *	
8 Comments		Subfiles (Total #: 1)	
1> subfile 1: X (512)	Start#	Shape	#Repts
2> timebase	1	1 x 512	2
3>			
4> subfile 2: Y (512) channel A			
5> Att: Offset:			
6> Probefactor:			
7>			
8>			
I/O Variable (R - Z): < Y >		.0000	
Subfile:	< 1 >	1 x 512 (Max Length: 1024)	
Row (0=all):	< 0 >		
Start Column:	< 1 >		
# of Columns:	< 512 >		
Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit: * Q *			

Note! depends on signal setting.

Screen 9.1

Select I/O variable X to copy the first subfile into ASYSTANT's variable X. Next, select subfile 1, press three times **Enter** to accept the displayed default values. There should now be a blinking cursor at the bottom of the screen. Select **(R)ead** to read the data from disk, wait until the data is read; this only takes a few moments. Move the cursor back to the I/O variable, select variable Y and subfile 2. Read the data from subfile 2.

(Q)uit and **Return**, to go back to the main menu of ASYSTANT.

Now data is ready for processing and/or display.

9.5. Display of data.

Before continuing with the calculation of the power spectrum, the data will be displayed.

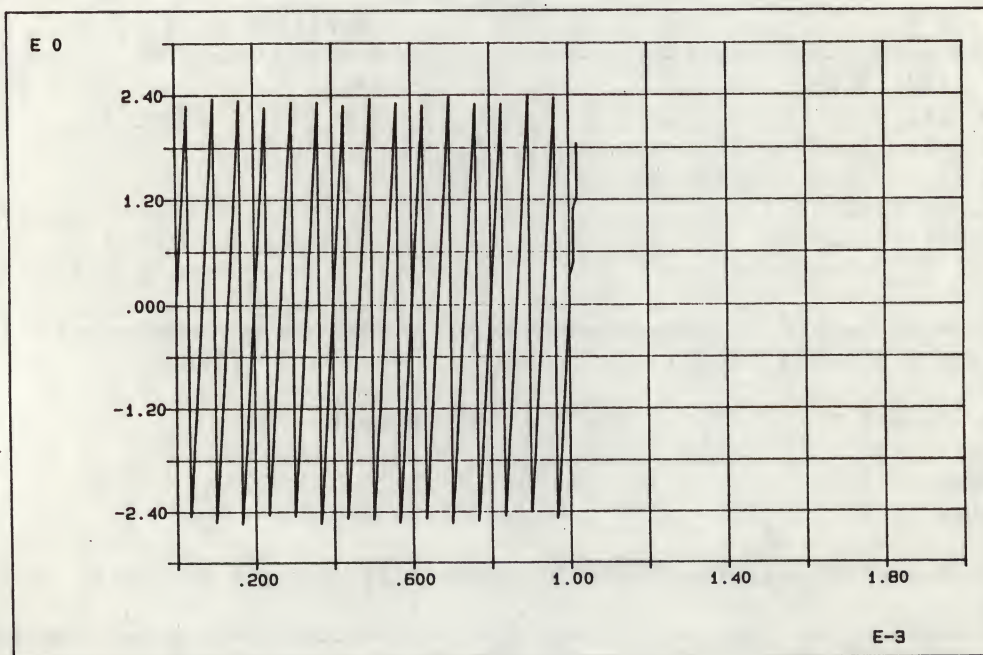
Select **Graphics** from the main menu options.

Select in the Graphics menu **XY AUTO** and specify:

x-array : X
y-array : Y

Press **<Esc>** en select **Display Graph**.

The picture from the oscilloscope is now displayed on screen 9.2 .



screen 9.2

Note! If you want to adjust the Y-axis, return to the main **Graphics** menu and select **Setup**. Define required **min** and **max** values of signal; press **<Esc>** to return to main **Graphics** menu.
Now select: **xy Axis**
 xy Plot

Return to the Main Menu by pressing the <End> and <Enter> keys, or move the cursor to option **Return**, and press <Enter>.

9.6. Calculation of the Power Spectrum.

The power spectrum is defined to be the squared magnitude of the Fourier transformation.

Put Y onto the stack (just press <Y> + <Enter>).
Press <PgUp> key to move the cursor to the desk calculator.
Enter the Wave and Matrix Operations menu by selecting next 3 times.

Note! In ASYSTANT options are selected when they are high-lighted.

Select in the Wave and Matrix Operations menu: **pwrspec** (= power spectrum).
Press <Enter> to start the calculation.

Almost immediately after pressing <Enter> the calculation is finished. The array on the stack is now called:

Unnamed Array.

This array, which contains the result of the power spectrum calculation, has to be stored.

Type: **STORE S**

Press the <PgDn> key and select the **Graphics Menu**.
Notice that the cursor remembers the previous selected option.
Select **y auto** and display the contents of variable S, showing the power spectrum.

One remark however: the displayed data is not scaled, and mirror frequencies are displayed.

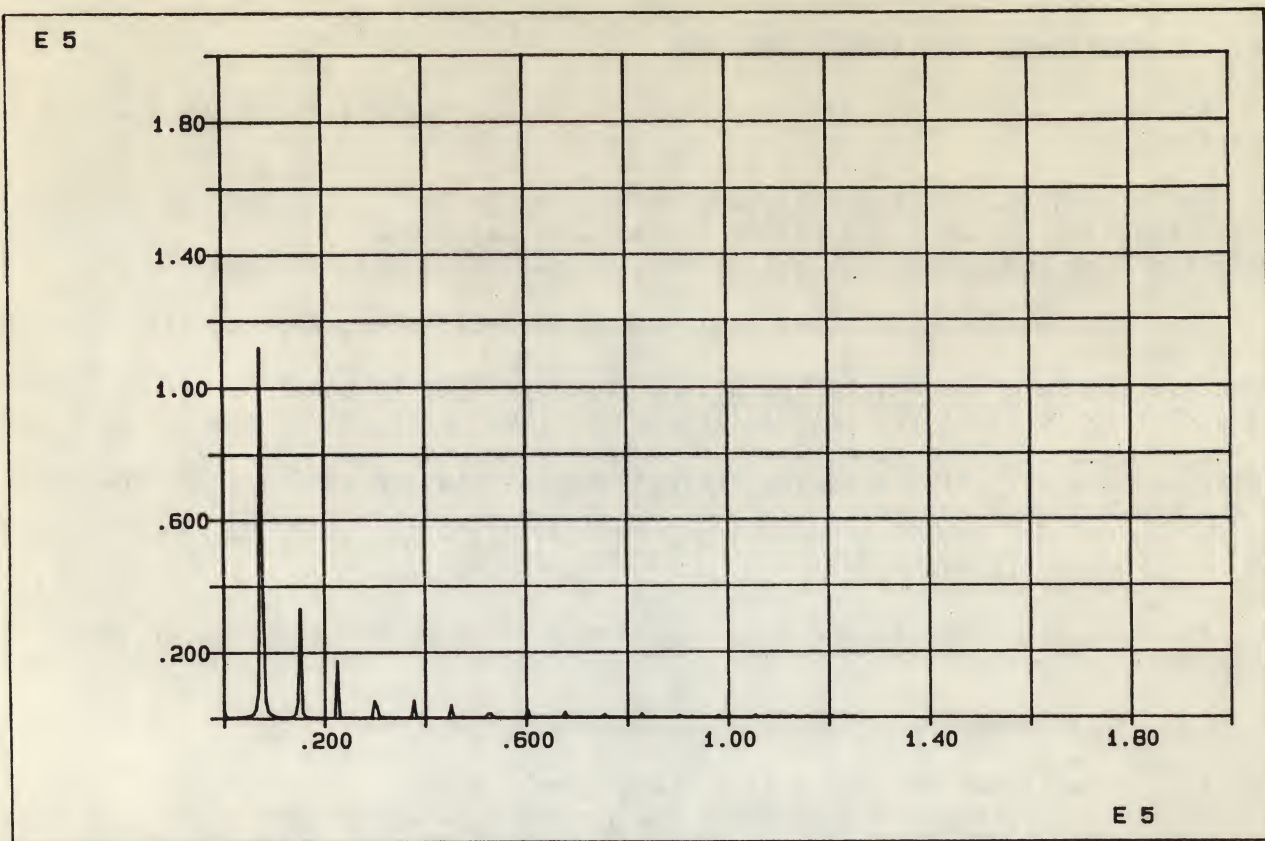
To scale the power spectrum, do the following:

Return to ASYSTANT's main menu. It is possible in ASYSTANT to define your own functions. The F1, F2, ... F10 function keys can each be assigned a user function, a sequence of up to 5 command lines. Specify the **Function** menu.

This user function calculates from a 512 point scaled array, which is stored in X and Y, a scaled power spectrum. The result is stored in R and S.
All you have to do is press function key F1.

X [512] STORE A	< 1 , B 2 / 1 + >
512 STORE B	STORE R
B A / STORE A	B N:RAMP 1 - B / A *
Y PWRSPEC	< 1 , B 2 / 1 + >
F2	STORE S

A scaled **power spectrum** is now being calculated and stored into I/O variables R and S. To display the spectrum, enter the Graphics Menu and select **xy auto**. Display the contents of I/O variables R and S.
The screen should now look as displayed in screen 9.3.



screen 9.3 : scaled power spectrum.

The last session is now finished. In principle you should be able to use the PM2260 software package for your own application. The remaining part of this manual can be used as a reference guide.



CHAPTER 1: **GENERAL FEATURES**

1.1 The ASYSTANT Environment

The ASYSTANT is an integrated system of advanced data analysis, graphics, and -- in the ASYSTANT+ version -- data acquisition functions.

ASYSTANT consists of a **Desk Calculator**, which provides much of the raw calculating capabilities of the system, and a number of **Main Menu** options. The Desk Calculator is the central hub of the system: data is passed between the options through Desk Calculator **Parameters** and **Variables**, which provide storage of numbers and arrays, and are always available for computation. These repositories are the primary mechanism for integrating the ASYSTANT'S diverse functionality.

The Desk Calculator and Main Menu are displayed on the same screen, and come up on entry after passing the initial setup prompts.

TIP

The tutorial sessions of this manual provide detailed, step-by-step instructions on how to set up and use the ASYSTANT. Work through these sessions at the keyboard first -- then come back to review the Reference chapters. They will be much more meaningful if you work with the system first.

1.1.2 THE DESK CALCULATOR

The Desk Calculator is designed to function like a pocket calculator, though it has many more capabilities. It consists of:

- * four menu pages of Calculator Functions
- * a command input line
- * three data storage areas
 - Stack (for temporary storage of scalars or arrays)
 - Parameter storage (for named scalars)
 - Variable storage (for named arrays or scalars).

Calculator computations are performed on the stack and data is transferred among the ASYSTANT functions through the Parameters and Variables. The Desk Calculator may be accessed from any Main Menu branch through a pop-up Mini Calculator and Variable display window. The Desk Calculator is described in detail in Chapter 2.

1.1.3 THE MAIN MENU

Advanced data analysis and I/O functions are selections on the Main Menu. If you have ASYSTANT+, the Acquire menu option is active and branches to an additional set of data acquisition menus. With the exception of Save/Exit, each option on the Main Menu branches to an ASYSTANT function and each function will return to the Main Menu on completion. The Main Menu options are discussed individually in Chapters 3 through 12. The Acquire option, and its function set, are covered in the Chapters 13 through 21 of the ASYSTANT+ manual.

1.1.4 MENUS AND PROMPT LISTS

The user operates ASYSTANT primarily through menus and prompt lists.

Menus are lists of options or choices. The current option is indicated by highlights or a pointer. (The pointer is used when graphics are displayed because highlighting is not provided by many video display cards in graphics mode). Pressing **Enter** causes the current option to be executed.

The arrow keys move the indicator through the menu list. The **Home** and **End** keys skip directly to the first and last prompts. To execute a menu option you must first move the highlight (or pointer) to that item, and then press **Enter**.

Prompt lists are used to enter values and in some instances, to select an action. A prompt list appears on the screen as a sequence of queries followed by default responses. The defaults are surrounded by either the characters < > or * *. A cursor is displayed beside one of the responses (initially the first). You may type in a new entry at the cursor location.

Each time you enter a new value the cursor will move on to the next prompt. You can use the arrow keys to move through prompt lists without changing default values, and the **Home** and **End** keys will take you directly to the first and last prompts.

Movement keys become inactive when the first character of a response is entered. Consequently, if you accidentally begin an entry, you must finish it with **Enter**; backspacing alone will not let you continue.

One of three basic types of response may be required in a given prompt: numbers, text, or single letters. The first two must be terminated with the **Enter** key. Single letters are generally grabbed directly from the keyboard. This may seem a little disconcerting at first, but eventually you will find that it reduces the number of keystrokes needed to carry out operations. In some instances, single letter prompts are used to initiate an action. These prompts are referred to as action prompts, and are distinguished on the display by the use of the delimiting characters * *.

The **Esc** key can be used to exit from prompt lists. It indicates to the system that you accept the current values and wish to continue. Often the last prompt will be a selection of what to do next, e.g., to quit or continue. If you press **Esc** in the middle of such a list, the system will still take the action specified by the default for the last prompt.

WARNING

If you get fouled up in the middle of a prompt list, look for a **Quit/Continue** prompt at the end of the list and select **Quit**. This will let you exit without performing any action. Keep in mind that **Esc** *accepts* the current state of the prompt list, and therefore is not a good choice if you want to quit or start over.

Some prompts specify file names. **ASYSTANT** will automatically modify entries into these prompts, if necessary, so that file names are uniquely specified by drive and extension. The previous entry in the prompt becomes the default for the next: if the drive is not specified in the new name, they will be taken from the old name. If neither the old nor the new file name has a drive specification, the temporary work drive that was declared in the system configuration process will be assumed.

1.1.5 SPECIAL KEYS

Several options are available "on-line" throughout the ASYSTANT. These can be invoked by pressing special keys, as listed below:

?	-- invokes the Help system
Space	-- Invokes the Mini Calculator
Ctrl-D	-- Calls up a menu of DOS operations
Ctrl-E	-- Exits the ASYSTANT and returns to DOS
Ctrl-F	-- Dumps the contents of the screen out to the Notepad
Ctrl-N	-- Calls up the Notepad
Ctrl-P	-- Dumps the contents of the screen out to the printer
Ctrl-V	-- Displays a list of the contents of the system Variables
Ctrl-Break	-- Returns you at any time to the Desk Calculator

More will be said about these operations in subsequent sections.

1.1.6 SAVING AND RESTORING

When you enter the ASYSTANT, it prompts you to recall a set of Variables, Parameters and Functions from a file. If you executed the Save option at the end of a previous session, you can restore this set and resume computation. However, the Save must be done explicitly when the system is exited or the information will be lost. ASYSTANT will prompt you to make a "save" file if you leave via Save/Exit from the Main Menu.

The save file is a specially formatted ASYSTANT data file. It will contain 68 comments and 18 arrays. The first 50 comments store the 50 command lines assigned to the 10 User Functions, and the other 18 comments store the Parameter and Variable labels. The 9 Parameters and 9 Variables are stored in the data arrays. Each Parameter is saved in an array of length 1.

WARNING

Save records only the Variables, Parameters, and Functions. The stack contents are not saved, nor is the current **status** (e.g., data type and format). See Chapter 2 for details on defining the system state.

1.1.7 MEMORY ALLOCATION

ASYSTANT uses two 64 kilobyte segments of memory for retaining data arrays. One segment is used exclusively for the R-Z system Variables, and the other is used to retain the temporary arrays that appear on the stack as "unnamed". No single array can exceed 64K bytes in size. Unnamed arrays are generated by many of the system's array operations. The space they occupy is freed as soon as they are removed from the stack. The unnamed array space is also used for carrying out operations like curve fitting, statistical computations,

etc. Therefore, if large unnamed arrays are left on the stack, it will limit the size of the arrays that can be handled by the Main Menu options.

If an operation requires more memory than is currently available, the error message:

Not enough transient space available

will be reported. Additional memory space must be freed up before the operation can be carried out. Either delete some data arrays or reduce their size. One possibility may be to temporarily copy them out to a disk file.

TIP

If you clear out all unneeded arrays, and you still are running out of memory, try the calculation with a lower precision array: single precision arrays take half the space of double precision. Be careful of mixing single and double-precision numbers in an expression (especially double-precision integers). The result will always be double-precision. For example, an expression: `2.33 2 +` will result in a double-precision real output, because the 2 (no decimal point) is interpreted as a double-precision integer. `2.33 2. +` takes up half the memory.

1.2 System Configuration

ASYSTANT will perform optimally if it is properly configured. You should spend some time to make sure that you are using the proper system configuration. There are basically four aspects to this: specifying your display hardware, setting up your disk drive(s), configuring a plotter, and configuring a printer.

Before you run ASYSTANT you should properly backup and install the ASYSTANT files. Please follow session 1 of the Tutorial to insure proper installation. As well, you must specify the correct DOS MODE command for your plotter. (See section 1.2.2 below).

When you load ASYSTANT, the initial prompt contains a **Setup** option. The **Setup** menu allows you to modify a series of default Parameters that are used to direct input and output, and to assign functions to disk drives. Whenever these Parameters are modified the new values are copied into the **ASYSTANT.CFG** file, which is automatically accessed each time the ASYSTANT is invoked. The **Setup** menu is displayed in the upper left corner of the screen and contains the following options:

- Display**-- Specifies the display hardware
- Plotter** -- Specifies the plotter
- Printer** -- Specifies the printer
- Files** -- Reassigns the disk drives used for system overlays, work files, and the "help" text.
- Return** -- Returns to the system entry prompt list

The current configuration is shown to the right of the menu, and is updated whenever a parameter is modified. In the following subsections the four aspects of configuration are discussed in more detail.

1.2.1 SPECIFYING THE DISPLAY HARDWARE

The Display option offers a choice of display hardware possibilities. It is very important to make the right selection here in order to utilize the maximum resolution and fullest range of colors of your display hardware.

Note: If you are using the standard IBM color graphics board, you will have color only in text mode displays. The color graphics available with this board provides a resolution too low for most scientific applications. It is preferable to sacrifice color in favor of improved resolution.

In Table R1.1 each display hardware option is listed, along with its resolution and color range. Also listed is the size of the file (in bytes) needed to save the "full screen" graphics window (see the **save** and **recall** options in reference chapter 4).

Hardware	Res	Colors	Image File Size
Enhanced Graphics Card (more than 64 Kbytes memory) Enhanced Color Monitor	640x350	16	96K bytes
Enhanced Graphics Card (more than 64 Kbytes memory) Standard Color Monitor	640x200	4	56K bytes
Enhanced Graphics Card (64 Kbytes memory) Enhanced Color Monitor	640x350	4	96K bytes
Enhanced Graphics Card (64 Kbytes memory) Standard Color Monitor	640x200	4	56K bytes
IBM Color Graphics Card Standard Color Monitor	640x200	B/W	16K bytes
IBM Color Graphics Card Black and White Monitor	640x200	B/W	16K bytes
Hercules Graphics Card Black and White Monitor	720x350	B/W	24K bytes
AT&T High Res Graphics Card and Monitor	640x400	B/W	32K bytes
HP Multimode Graphics Card and Monitor	640x400	B/W	32K bytes
Enhanced Graphics Card Monochrome Monitor	640x350	B/W	96K bytes

TABLE R1.1 *Display Hardware Options*

The display setup also includes an option for turning off graphics colors, allowing screen dumps of graphic displays to be faithfully generated. Screen dumps made (using Ctrl-P) with colors turned on may not reproduce some features appearing on the screen. This is due to an unavoidable confusion in mapping the different display colors into black and white.

1.2.2 CONFIGURING A PLOTTER

To use a digital plotter, it is necessary to connect the COM1 serial port on the PC to the plotter. The plotter must have the RS-232 serial port option.

Once the plotter is connected, it is also necessary to use the right DOS MODE command to set the baud rate, parity, number of data bits and number of stop bits to match the setting on the plotter. These settings are established by dip switches located on the back of the plotter. For example, to set up the HP7470A plotter, you might use the following switch settings:

Switch	Setting	Significance
S2	0 	No Parity Checking
S1	0 	
Y	D	Direct link to computer
US	1	
B4	1 	Baud rate 4800
B3	0 	
B2	0 	
B1	1 	

The MODE command corresponding to these settings is

MODE COM1:48,N,8,1,P

If a plotter is going to be used routinely, you might want to install the MODE command in your AUTOEXEC.BAT file so that it will always be executed when you power up the system. (Refer to the DOS manual for more information on the MODE command and .BAT files.)

Finally, ASYSTANT must know the type of plotter you have. This information is specified by executing the **Plotter** option in the **Reconfigure** menu. This will be saved in the ASYSTANT.CFG file and becomes the default for subsequent sessions. ASYSTANT will recognize three types of plotters: HP7440, HP7470, and HP7475. Other plotters which are HPGL compatible, such as the Gould Colorwriter, may be compatible as well.

1.2.3 CONFIGURING A PRINTER

It is necessary to configure a printer only if you wish to use the **Ctrl-P** option to generate screen dumps of graphics displays. The PC's built in **Shift-PrtSc** will not work in the graphics mode in many display hardware configurations. **Ctrl-P**, on the other hand, will work with any of the supported display configurations, and is a "legal" keystroke for Prokey type programs.

Printer selection is made by executing the **Printer** option in the **Reconfigure** menu.

If your printer is not listed, it may be compatible with one that is, so try out each of them in turn, starting with the **IBM Graphics Printer**.

Whenever a new printer is selected the **PRINTER.TBL** file is accessed, and 3 configuration tables are copied from this file into the **ASYSTANT.CFG** file. Three tables are needed for each printer because a different printer configuration is required for each horizontal display resolution (i.e., 200, 350, and 400). The **PRINTER.TBL** file must be available on the same drive as the **ASYSTANT.CFG** file in order to reconfigure the printer.

1.2.4 SETTING UP YOUR DISK FILES

ASYSTANT uses several different types of disk files and it is helpful to be aware of how it uses them. ASYSTANT is a comprehensive software product and requires a fair amount of disk space to house all its options. If you have a two-floppy system it is especially important to organize these files carefully. A general description of ASYSTANT disk files and disk usage is given below. Disk configuration is discussed at the end of the section.

The following is a list of the disk files that come with the ASYSTANT:

ASYSTANT.COM	-- Main System COM file
ASYSTANT.CFG	-- Configuration file
ASYSTANT.MSG	-- Error Message file
ASYSTANT.HLP	-- Help System file
ASYSTANT.OVL	-- Main System Overlay
CFG.SYS	-- Special Configuration Overlay
CURVFIT.OVR	-- Curve Fitting Overlay
DIFFEQS.OVR	-- Differential Equations Overlay
FILEIO.OVR	-- File I/O Overlay
FILEPROC.OVR	-- Multi Processing Overlay
GRAPHICS.OVR	-- Graphics Overlay
POLYOPS.OVR	-- Polynomial Operations Overlay
STATOPS1.OVR	-- Statistics Overlay 1
STATOPS2.OVR	-- Statistics Overlay 2
WAVGEN.OVR	-- Wave Generation Overlay
WAVPROC.OVR	-- Wave Processing Overlay
PRINTER.TBL	-- Printer Table File

Plus sample data files.

If you have purchased the ASYSTANT+ then you should also have the files:

ACQUIRE@.CFG	-- Acquisition Hardware Configuration file
CONVERT.OVR	-- Data File Conversion Overlay
CFGACQ.OVR	-- Function Generator Overlay
DLPE.OVR	-- Data Logger Overlay 1
DLAQ.OVR	-- Data Logger Overlay 2
FG.OVR	-- Function Generator Overlay
HS.OVR	-- High Speed Recorder Overlay
HWS.OVR	-- Hardware Scroller Overlay
SA.OVR	-- Signal Averager Overlay
SC.OVR	-- Strip-Chart Recorder Overlay
THRMCPPL.NBS	-- Thermocouple Linearization Coefficients
TR.OVR	-- Transient Recorder Overlay
XY.OVR	-- XY-Recorder Overlay

Plus sample data and .DFP (data file parameter) files.

The ASYSTANT+ files are discussed in chapter 13.

1.2.5 MAIN SYSTEM AND CONFIGURATION FILES

ASYSTANT.COM and ASYSTANT.OVL are the two main system files and must both be available every time the system is loaded. These files are found on the System and Key Diskettes, respectively. The .COM file must be resident on the logged disk in order to call up the system from DOS. The .OVL file is automatically read each time the .COM file is executed, but if it is not found on the logged drive it will be automatically sought on all other drives. CFG.SYS and ASYSTANT.CFG are also always read when the system is booted and should reside on the same disk drive as the .COM file. The CFG.SYS file is a special overlay used to configure the system and ASYSTANT.CFG is a data file containing default configuration Parameters. All configuration files are found on the System Diskette.

The two main system files and the CFG.SYS are no longer needed once the system is loaded and the initial Recall/Setup/Continue prompt appears. If Setup is selected, the ASYSTANT.CFG file must be available until the configuration is complete and you return from the Setup menu. Thereafter, ASYSTANT.CFG is not needed until you load ASYSTANT again. In sum, once Continue is selected, the system and configuration files are no longer necessary and the System and Key Diskettes can be removed from the drives.

1.2.6 OVERLAY FILES

All files with the extension **.OVR** are system overlays called up by options from the Main Menu. **ASYSTANT** option overlays are found on the Analysis Diskette; **ASYSTANT+** option overlays are on the Acquire 1 and 2 disks. Each overlay is read only when the corresponding menu option is invoked, so if you rarely use some menu options, you may wish to omit corresponding **.OVR** files from your working copy of the overlay diskette(s). The **Setup** menu allows you to select the system drive that will be read each time an overlay is called up.

1.2.7 ERROR MESSAGE FILE

The file **ASYSTANT.MSG** contains error messages. It is read each time an error is encountered and also when the system is booted. It must be available in the system overlay drive each time the software is loaded, but it does not have to remain available throughout the session. However, if an error is encountered and the **.MSG** file is not present, then the error will be reported without an explanatory message.

1.2.8 THE HELP FILE

The file **ASYSTANT.HLP** contains the text for the on-line help system. The **Setup** menu allows you to select the system drive that will be used for this file. Each time help is requested the **ASYSTANT.HLP** file will be accessed on that drive.

1.2.9 SYSTEM WORK FILES

It is sometimes necessary for the system to generate transient files to either retain screen images or to carry out various other system functions. The names of these work files are listed below.

VUPORT.TMP
ERR.TMP

Generally, these files are automatically erased each time they are used, but if for some reason the system is interrupted before the erasure, they may then appear in your directory. These work files can be directed to any disk drive via the **Setup** menu. The sizes of these files will be related to your display hardware (See Table R1.1).

1.2.10 USER FILES

In addition to the files listed above, the ASYSTANT can generate a number of different user files. The various types are listed below.

- | | |
|---------------------|---|
| ASYSTANT Data Files | -- The standard ASYSTANT data file is a binary file containing data arrays and comments. These files can also be converted to or from DIF or ASCII formats via the File I/O menu. |
| Note Files | -- Note files are standard text files that can be edited by most common editors. They are produced by the systems Notepad option (see section 1.4). |
| Graphic Images | -- These are binary files that are used to retain the pixels of a graphic display. The file size is determined by the size of the saved window and the display hardware (see Table R1.1). |
| Parms/Vars/Functs | -- Each time the system is exited the contents of the system Parameters, Variables, and Functions can be saved so that they can be recalled at a later time. They are saved in a standard ASYSTANT data file. |

Each of these files can be read from or written to any system drive.

1.2.11 HARD-DISK CONFIGURATION

If you have a hard disk then you will probably want to keep all system files on it, including the overlay, error, and help files. Execute the Files option in the Setup menu and specify the drive of the hard disk for the overlay, work, and help files.

1.2.12 TWO FLOPPY CONFIGURATION

The five diskettes that come with the ASYSTANT (seven for the ASYSTANT+) are already set up to be appropriate for use on a two-floppy system. They leave little remaining space for any user files, but this is not really a problem because the System and Key Diskettes can be removed and replaced with an overlay disk and a data disk as soon as the system is loaded and configured. In this case it is generally best to select one drive for overlays and the other drive for work files and help.

TIP

To minimize diskette shuffling, make up your own diskette that contains all of the overlays that you commonly use, plus the essential setup files. For example, if you commonly work with the high speed recorder, File conversion, File I/O, and Waveproc, make up a diskette containing:

HS.OVR
CONVERT.OVR
FILEIO.OVR
WAVEPROC.OVR
ASYSTANT.COM
ASYSTANT.MSG
ASYSTANT.CFG
CFG.SYS
ACQUIRE@.CFG

Use that in your A: drive instead of the distribution System and overlay diskettes.

If all of the overlays that you wish to have readily available cannot fit onto one diskette along with the setup files, leave the setup files off (but keep ASYSTANT.MSG if you want error messages displayed). You will need to use the System Diskette to load ASYSTANT+, but can then replace it with the overlay disk once you are in the system. Thereafter, you will only need to reinsert the System Diskette if you select the **Configure** option in the **Acquire** menu.

1.3 Using Help

This menu contains options for stack manipulations, standard elementary functions, and also several miscellaneous operations like store, prnt, pedit and status. The functions all act on the entry on the top of the stack, and work with all data types including complex numbers. However, radian angular units must be in effect for functions to work properly on complex arguments. When a function is applied to an array, the result is a new unnamed array of the same shape as the original. The values of the resultant array are obtained by applying the function to the original array on an element by element basis.

next

Advances to the next menu: Array Operations

store

Type space bar to continue...

Current: Desk Calculator

⌘F10 - Exit

F1 Back to: Asystant

F2 Calculator Overview

F7 Conversions Menu

F3 Entering Numbers

F8 Using the Command Mode

F4 Basic Arithmetic

F5 Desk Calculator Menu

F6 Array Operations Menu

SCREEN R1.1

The Help system has been designed for quick access to information about ASYSTANT menus. Help may be entered by typing "?" from any menu or prompt list, and provides a display similar to Screen R1.1.

Note: If you enter Help from a graphics display mode, ASYSTANT must first store the current video display on the drive specified for temporary work files in a file called VUPORT.TMP. Depending on your display resolution, this image file may be as large as 96K bytes (see Table R1.1) and storing the image may take several seconds.

At entry Help displays information about the ASYSTANT menu from which it is called. For example, if you are in the **Graphics** menu, Help will display information about Graphics. Help text is based on the Reference Section of the ASYSTANT manual.

The Help system is organized using a tree format. You can move down the tree to more specific information about the current menu or up the tree to information about entirely different menus. Related menu items are grouped together.

When you enter Help, the screen is split into two areas (see Screen R1.1). On the upper half, text appears. On the lower half is a list of function keys and prompts for menus or concepts. The F1 key always allows you to move up the tree one level. By pressing any other of the listed function keys, you can move down the tree to the next level or off into a different branch. Each step up or down the tree will produce a new screen of text and a new list of function key options.

On occasion, the text associated with a particular branch may be too long to list in one screen. In these instances, the listing will pause. The space bar will cause the listing to continue, while a function key will stop the listing and execute the function key option.

Some function key listings show an underlined "a" in front of them. This indicates that you must hold down the Alt key while pressing the corresponding function key.

To leave Help at any point, simply press **Alt-F10**. You will return to the same menu from which you called Help. Note that if this menu is in a graphics display mode, then VUPORT.TMP file will be read back from the work disk to re-paint the screen.

Help requires the disk file ASYSTANT.HLP. This must reside on the same drive as your menu overlays. If ASYSTANT cannot find ASYSTANT.HLP, it will report an error and return you to the menu from which you invoked Help. Two-floppy owners should therefore insert the Help disk before pressing "?". Remember to remove the Help Disk and return the original disk to the drive when you exit the Help system.

1.4 The Notepad Editor

The ASYSTANT notepad editor allows editing of text and numerical output from within the ASYSTANT's menu driven environment. The notepad editor is designed to be simple to learn; it is not meant to replace a word processor for sophisticated text formatting. The ASYSTANT editor has three format restrictions:

1. The file must be a standard ASCII DOS file ending with an <EOF> character (^Z). (Note: WORDSTAR files created in the "D" mode do not conform to this rule and are incompatible. However, WORDSTAR files created in the "N" mode are acceptable).
2. The file must be smaller than 16 Kbytes.
3. The file may not have any lines that exceed 80 characters in length.

The ASYSTANT editor cannot read or create any file that violates these restrictions.

1.4.1 THE OUTER EDIT MENU

The notepad editor is entered by typing **Ctrl-N** from any menu within the ASYSTANT. If you enter the editor from a graphics display mode, ASYSTANT has to store the current video display on your work disk in a file called VUPORT.TMP. Depending on the display resolution, this image file may be as large as 96K bytes (see Table R1.1 in section 1.2.1) and storage may take several seconds. Once you enter the editor, the **outer menu** screen appears [refer to Screen R1.2].

Default file is: C:NOTES.DOC Type file specification or <cr>:	
F9 Directory F10 Exit	

SCREEN R1.2

The F9 key will display a directory of any disk in the system after prompting for a directory specification. Valid responses include:

- * A drive specification such as A:, B:, C:, etc.
- * A global file specification such as A:*.DMO
- * A specific filename such as MYNOTES.DOC

The directory listing may be paused at any time by pressing any key; the listing is continued by pressing a key again. Pressing any key three times in rapid succession will interrupt the listing and return you to the outer menu. Once the directory is listed, the file specification prompt returns to the screen.

The F10 key (exit) will return to the menu from which the editor was invoked.

To create a note file or edit an existing file, enter the file name at the **filename** prompt using the standard DOS syntax of: d:filename.ext (drive:filename.extension). If the drive, filename, or file extension is omitted, the displayed default will be used. Pressing **Enter** alone accepts the complete default file listed (the default file is NOTES.DOC on initial entry). When a valid file name is entered, the screen is cleared and repainted with a **text window** and the **edit menu**.

TIP

If you do not wish to append a file extension to the filename, terminate the name with a period (.), e.g., :

d:MYFILE.

Otherwise, the default extension will be used.

If the **movement-increment object** is **Character** (the default) the cursor movement keys behave as follows. The **Home** key returns the cursor to Row 0 Col 0 of the current screen. It does not return to the top of the file. The **End** key similarly returns the cursor to Row 19 Col 0 on the screen. The **PgUp** key scrolls full pages of twenty lines towards the top of the file while the **PgDn** key scrolls pages toward the end of the file. The four **arrow** keys move the edit cursor anywhere on the current screen and scroll the text off the current screen one line at a time.

F1 will change the object to **Word**. Subsequent use of the **right arrow** key will move the cursor one word to the right each time the key is pressed. When you advance to the last word of a line, pressing the **right arrow** one more time will advance the cursor to Col 79. The current object will remain **Word** until either another object key or the **up** or **down arrow** key is typed. This will return the editor to the default **Char** object.

F2 (Line) allows movement to the the start and end of a line with one stroke of the appropriate **arrow** key. **F3 (Page)**, used with the **up** and **down arrow** keys, is effectively equivalent to **PgUp** and **PgDn**. The **F4 key (File)** combined with either the **up** and **down arrow** or **Home** and **End** will move the cursor to the top and bottom of the notepad file.

The status line will always show the current object. In general, once a **Line**, **Page**, or **File** movement is made, the status line will return to the **Char** object. This means that if you move to the top of the file by pressing **F4** then the **Home** key (file top), you must press **F4** again and then press **End** to move to the end of the file.

Table R1.2 at the end of this section describes these and additional uses of the cursor movement and function keys.

1.4.3 TEXT DELETION

The direction arrow has significance only when deleting text. **F5 (direction)**, will toggle the direction arrow on the status line between **-->** (right) and **<--** (left). If the direction arrow points to the right, the current object is **Char**, and the editor is in **Insert** mode, pressing **Del** will cause the character at the cursor location to be deleted and all text to the right of the cursor moved left one position. If, in the same context, the direction arrow points to the left, the character to the left of the cursor is deleted and the line is "closed up".

Deletion while in **Overwrite** mode is similar except that characters are replaced by spaces.

The object keys can be used to delete a word, line, page, or file. For example, pressing **F1** followed by **F5** (reversing the direction arrow), and then **Del** will delete the word to the left of the current cursor position. To prevent accidental erasure, the **Shift** key must be pressed in conjunction with the page (**F3**) and file (**F4**) object keys in order to delete text. A **Delete** message will appear on the status line in this case. (E.g., **Shift-F4 F5 Del** will delete from the current cursor position to the top of the file).

1.4.4 SAVING A MODIFIED FILE

Any time the file has been modified, a message "File changed since save" will be displayed directly beneath the status line. Changes may be saved at any time by pressing F9 (Save). When a file is saved, ASYSTANT first checks to see if there is a file on the diskette with the extension .BAK. If there is, the editor deletes it. Next, the entire current file, which is completely memory resident, is written to a file with the extension .\$\$\$. This ensures there is adequate space on the disk to save the edited file, before ASYSTANT closes the old file. If this is successful, the editor renames the original file to the extension .BAK. Finally, the editor renames the temporary .\$\$\$ file to the original file extension.

1.4.5 TEXT SEARCHING AND REPLACEMENT

Searching for and replacing text can be done using F6. Searching is always performed from the current cursor position toward the end of the file; it is not possible to search backwards. When F6 is pressed, three choices will be displayed:

- * F: Find
- * R: Replace
- * A: Again

Thus the sequence F6 F can be used to search the file for a string. The editor will prompt you for a target string. If there is more than one occurrence, the sequence F6 A will search forward to the next occurrence of the same string. This can be repeated as many times as necessary to find the desired string.

Similarly, any string may be replaced with another by the sequence of F6 R. In this case, the editor will prompt for both a target and replacement string. If the string is found, it will be replaced; there is no query option. Subsequently selecting the Again option will repeat the search and replace operation.

1.4.6 MOVING BLOCKS OF TEXT

Blocks of text within a file can be copied, moved, and/or deleted. Such blocks are limited to whole lines of text; fragments of lines cannot be defined. To specify a block, markers must be set to indicate where the block begins and ends. Both markers are set using F7. The beginning marker is set by moving the edit cursor anywhere on the first line of the text block and typing F7. This will highlight the entire text line. At this point, the text cursor can be moved to the last line in the text block. F7 will highlight the entire block of text between the two marks. Once the beginning marker is established, the end marker may be set many times, shrinking or expanding the marked block, as long as it remains after the beginning marker. If you attempt to set the end marker ahead of the beginning marker, you will receive an error message.

Whenever a valid block is marked, the **Block oper** option on the edit menu will be highlighted. There are four block operations:

- * C: copy
- * M: move
- * D: delete
- * U: unmark the block

If you wish to delete the marked block, simply press **F8 D**. The text within the marked block will be deleted from the file, the screen will be updated and both text markers removed.

To move a marked block of text, place the text cursor at the destination point and type **F8 M**. The text will be removed from its previous location and placed at the current cursor location.

To copy a marked block, move the cursor to the destination point and type **F8 C**. The text will be replicated at the current cursor location.

The block operations will always insert the marked block ahead of the current cursor line. Note that the destination of a block copy or move can be anywhere in the file.

After copying or moving, a block will remain highlighted. This allows you to keep moving the same block until it is properly positioned or to copy a block many times at different locations in the text. To copy a block a second time, just move the cursor to the appropriate position for the additional copy and press **F8** again.

Note that when any text in the file is modified, the block markers will be removed. Similarly, pressing **F8 U** will remove the current markers from a defined block.

1.4.7 EXITING AN EDITING SESSION

Changes can be saved at any point without leaving the edit menu by pressing **F9** (save). You will notice that the message "File changed since save" no longer appears below the status line. If you attempt to exit an editing session without first saving your changes the notepad editor will prompt you to save the file. Responding with a **Y** at this point will save editing changes. Of course, there are times when you may not desire to save changes. In this case, respond with an **N** and ASYSTANT will exit the editor without saving the changes.

After pressing the **F10** key (Exit), the outer menu will editing, a directory listing can be shown (**F9**), or you can exit (**F10**) to the ASYSTANT menu from which you invoked the editor.

Function Key Assignments

Outer Menu
F9 Directory
F10 Exit

Edit Menu
Movement Increment Object
F1 Word
F3 Page
F4 File

Deletion Direction Arrow
F5 Direction

String Operations
F6 Find&Repl

Block Operations
F7 Block Mark
F8 Block Oper

Save and/or Exit
F9 Save
F10 Exit

Useful Key Combinations

F1 <-	go to word left
F1 ->	go to word right
F2 <-	go to first char on line
F2 ->	go to last char on line
F3 <-	go to lefthand margin of screen
F4 Home	go to top of file
F4 End	go to end of file
Shift-F3 F5 Del	delete to top of screen
Shift-F4 F5 Del	delete to top of file
Shift-F3 Del	delete to end of screen
Shift-F4 Del	delete to end of file
<Return>	split line at cursor position
Cntl-<Return>	concatenate lines

TABLE R1.2 *Editor Function Key Assignments and Combinations*

1.5 Using DOS Operations

To help you maintain disk files within the ASYSTANT a set of DOS operations are available: file deletion, file renaming, file copying, and file directories. **Ctrl-D** calls the menu of DOS Operations from inside any other menu or prompt list. The **Return** option in the DOS Operations menu returns you to the menu or prompt list from which you came.

An unambiguous specification of a file name requires a drive, file name, and extension, written as "d:filename.ext". To ease the specification of file names, the last specified drive is used by default. On initial entry, the drive defaults to the Temporary Work File drive selected during the setup of ASYSTANT. It is strongly recommended, however, that you use complete file specifications to avoid file disasters. Note also that ASYSTANT does not support pathnames to other subdirectories.

A summary of the DOS capabilities within the ASYSTANT are given next. It is extremely important to note that **you must not delete the file VUPOINT.TMP** from the current work disk when using the **Ctrl-D** option. This file is created by the ASYSTANT for temporary storage of the graphics picture, and will be deleted automatically at the proper time.

Delete File

Opens a prompt list to specify the name of the file to be deleted. The wild card symbols "*" and "?" can be used. A **Quit/Continue** prompt is available to allow you to **Quit**, and return to the DOS menu, or **Continue**, and delete the file(s).

Copy File

Opens a prompt list to specify the names of the source and destination file. Both file names must be specified, even if the name is the same and only the drive is different. The wild card symbols "*" and "?" *cannot* be used. A **Quit/Continue** prompt is available to allow you to **Quit**, and return to the DOS menu, or **Continue**, and copy the file.

Rename File

Opens a prompt list to specify the names of the source and destination files. The wild card symbols "*" and "?" *cannot* be used. A **Quit/Continue** prompt is available to allow you to **Quit**, and return to the DOS menu, or **Continue**, and rename the file.

Directory

Opens a prompt list to specify the file(s) to be listed. The wildcard symbols "*" and "?" can be used. A **Quit/Continue** prompt is available to allow you to **Quit**, and return to the DOS menu, or **Continue**, and list the file(s). You may pause and then continue the listing by pressing the space bar. You may also abort the listing at any time by pressing the space bar three times in rapid succession.

1.6 The Mini-Calculator

The Mini-Calculator is a facility available in all Main Menu options. It is used to carry out short computations or to modify system Variables and Parameters without having to return to the Desk Calculator. It is invoked by pressing the **space bar**. The only times that the Mini-Calculator cannot be used are:

1. In the Desk Calculator itself;
2. While editing in the Notepad, or carrying out DOS operations;
3. When entering text (e.g., entering user functions, editing file comments, etc.).

In the Mini-Calculator, operations are specified exclusively on the command line; the menus of calculator functions are not displayed. (See Chapter 2 for a detailed description of the command line. Section 2.10 covers the particulars of using the command line as an alternative to menu selection.) When the **space bar** is pressed, the Mini-Calculator appears in a window on the screen. The stack is in the upper left-hand corner of the window, with a command line window just beneath it. Each time a command is carried out, the stack is refreshed and the command line is erased. The Mini-Calculator is exited by typing **DONE** and pressing the **Enter** key.

When the Mini-Calculator is called from a graphics display mode, the contents of the display window that it displaces are copied into the disk file VUPORT.TMP. This file is then used to restore the display window when the Mini-Calculator is exited.

1.7 Error Handling

A common error handling system is used throughout the ASYSTANT. Each error is coded with a number that is used as an index into the message file ASYSTANT.MSG. Thus, when an error is encountered, the appropriate message is read from the ASYSTANT.MSG file and typed out on the display. This file must be on the system overlay drive when the error occurs. If file access is unsuccessful, the error number will be reported on the display without the associated message. The error numbers and their corresponding messages are listed in Appendix B.

If an error occurs while the system is in a graphics display mode, then the contents of the window used for reporting the error will be copied into the file ERR.TMP on the work disk before the error message is displayed. But in the event that it is impossible to save this display image (for example, if the disk drive is open or if the disk is full), then the error will be reported along with another message stating that the current display contents have been lost.

You can recover from most errors by pressing any key. This generally causes the system to resume processing at the menu option or prompt selection that was nearest to the point where the error was encountered. In some cases it may be necessary to restart the system by pressing **Ctrl-Break**. This will not result in the loss of any data, though the data may have already been corrupted by the error.

There are two additional points to keep in mind when replying to ASYSTANT errors.

1. It is often the case that one or more of the stack entries were corrupted by the error, especially if the error occurred during the execution of a calculator function or command line entry. Many operations use the stack to retain intermediate values, so if an error occurs in the middle of a complex operation then some of the stack contents may be spurious.
2. Some errors messages may not seem to directly relate to the operation being performed, however, they probably refer to difficulties that are encountered in intermediate operations.

CHAPTER 2: **DESK CALCULATOR**

Stack Contents					Parameters	
W	REAL DIM[2 , 22]				COST	A= 2500
Y	DP.INTEGER DIM[4 , 4]				UNITS	B= 13
	DP.REAL DIM[5]					C= .0000
						D= .0000
						E= .0000
						F= .0000
						G= .0000
						H= .0000
						I= .0000
Calculator Functions					Variables	
next	store	pedit	status	print		
dup	drop	swap	roll	pi		
sin	asin	sinh	asinh	inv		
cos	acos	cosh	acosh	neg		
tan	atan	tanh	atanh	abs		
exp	ln	10exp	log	sqrt		
Main Menu Options					INPUT	R= 100 REAL
Acquire	Graphics	Functions				S= 100 REAL
File Proc	File I/O	Polys				T= .0175
Wave Proc	Curve Fit	Diff Eqs				U= .0000
Wave Gen	Stats	Save/Exit				V= .0000
						W= 2 x 22 REAL
						X= .0000
						Y= 4 x 4 DP-INT
						Z= .0000

2.1 Overview

2.1.1 GENERAL DESCRIPTION

The Desk Calculator is the central data processing and storage hub of the ASYSTANT. It provides a powerful computational tool in the familiar form of a desktop calculator and is also the means of passing data to ASYSTANT's feature functions. The **PgUp** and **PgDn** keys can be used at any time to toggle between the Desk Calculator menus and the Main Menu of features.

The calculator performs operations on entries which have been stacked in memory. These entries can be one or two dimensional arrays as well as single numbers, which makes the ASYSTANT far more powerful than any ordinary calculator. It will also recognize complex numbers, and has many built-in special functions such as Bessel, Error, and Gamma functions. These

functions, like all the others, can be applied to both real and complex numbers, and even to entire arrays. The Desk Calculator also provides sophisticated array and waveform operations, such as sorting, matrix inversion, and fast Fourier transformations, which can be executed through a single keystroke.

The Desk Calculator uses five windows for its display: **Stack Contents**, **Parameters**, **Variables**, the menus of **Calculator Functions**, and a **Command** input line which is displayed when any number, letter, or arithmetic operator key is pressed.

2.1.2 THE CALCULATOR MENUS

The total set of calculator functions spans four menus which are displayed in the center left window of the Desk Calculator. The **next** menu option appearing in the upper left hand corner of each menu is used to page from one menu to the next.

The first, called **Calculator Functions**, contains the set of basic functions and stack operations that are common to most scientific calculators. Basic arithmetic operations (+, -, *, / and ^) are not included in this menu. These operations are carried out on the **Command** input line by pressing the appropriate key. Also on this menu are two special options for handling calculator entries. These are **store**, which stores stack entries in **Parameters** or **Variables** much like a memory button, and **status**, which allows the user to select data type formats such as integer, real, or complex, double or single precision, radian, degree, or gradient, etc.

The second menu is titled **Array Operations**. It contains an extensive set of operations for generating and manipulating arrays. The ASYSTANT supports both one and two dimensional arrays of up to 64 Kbytes in length. The maximum number of elements depends on data type. (See 2.2.3). The **aedit** (array edit) option in this menu provides an easy means of creating arrays and entering array elements one at a time.

The third menu contains **Conversions and Special Functions**. The options in this menu allow for real/complex, polar/cartesian, and spherical/cartesian conversions. This menu also includes a set of special functions useful in advanced applications.

The fourth and final menu displays a selection of **Wave and Matrix Operations**. Operations that act on waveforms, such as **fft** (fast Fourier transformation), **smooth**, and **autocorr** (auto correlation), as well as matrix operations such as **determ** (determinants), are found on this menu.

2.1.3 THE COMMAND INPUT LINE

Numbers and simple algebraic operators are entered into the Desk Calculator on the command input line. When any number, letter, or operation key (+, -, /, *, ^) is pressed the command line window appears at the bottom of the Desk Calculator display. When the **Enter** key is pressed the entries are pushed onto

the stack and/or the operations are performed. Commands may be entered in Reverse Polish Notation (RPN) or algebraic syntax. As well, most options on the calculator menus may be entered at the command line as an alternative to menu selection. Commands may be entered in any combination of upper and lower case.

2.1.4 THE STACK

The Desk Calculator stack is an area of memory used for computations and temporary storage of numbers and arrays. Numbers entered into the calculator are "stacked" in the stack memory. The ASYSTANT stack accepts entries by the last-in/first-out (LIFO) method, also known as the "push-down" method. When a number is entered on the stack, it is "pushed" onto the top of the stack and when an entry is removed, it is "popped" off the top of the stack, much like a stack of dinner plates in a cafeteria. Operations are performed on stack entries using Reverse Polish Notation (RPN). Operations pop the top entry or entries off the stack, perform the calculation and push the result back on the stack. The stack holds up to five entries which are displayed in the **Stack Contents** window at the top of the Desk Calculator screen.

2.1.5 PARAMETERS AND VARIABLES

In addition to the stack, data can be retained in the nine Parameters A-I, and in the nine Variables R-Z, displayed in the right hand windows of the Desk Calculator screen. Parameters differ from the Variables in that they can only retain numbers, while Variables can store both numbers and arrays. The contents of a Parameter or Variable can be copied to the top of stack by pressing the associated letter key followed by the Enter key (i.e., entering the letter on the command line). The **store** function is used to pop the top stack entry into either a Parameter or Variable.

Besides being a computational convenience, the Variable and Parameter data stores are used to pass data to the Main Menu options such as **Graphics** or **Statistics**.

2.1.6 USER FUNCTIONS AND POLYNOMIALS

Two other Desk Calculator features are user defined functions ("macros") and polynomials. User Functions are sequences of operations which are assigned to the function keys F1-F10 (see Chapter 3). The functions are automatically executed when the associated function key is pressed, or alternatively, when spelled out on the command line (eg., F1). Also, up to ten polynomials can be defined (see Chapter 10) and applied as functions to the top stack entry by entering **Pn** on the command line, where n is the number, 1-10, of the polynomial.

2.2 Entering Numbers

2.2.1 PUTTING NUMBERS ON THE STACK

Numbers can be placed on the top of the stack by simply typing them in at the keyboard. The command window overlays the Main Menu at the bottom of the screen containing the text:

Command: <1st-digit>__

The remaining digits can then be entered, and the numeric input terminated by pressing the Enter key. Once the number has been entered, it will immediately appear in the stack display in the upper left hand corner of the screen. Note that several numbers can be entered in the same command line provided that they are separated from each other by spaces. As successive numbers are placed on the stack, their predecessors are pushed down deeper and deeper. When the stack is full, pushing new entries on the top of stack will cause the old ones to be deleted from the bottom.

Real numbers must be entered with a decimal point. Real numbers will default to single-precision unless terminated with the letter "D" for double precision (eg., 3.2D).

Conversely, integers default to double-precision, unless terminated by the letter "S" (eg., 3S).

2.2.2 ENTERING COMPLEX NUMBERS

Complex numbers can be entered and displayed in either cartesian or polar form (see section 2.6, **status**). Internally, however, they are always stored in cartesian form. The third calculator menu, Conversions and Special Functions, gives options for converting data forms (see section 2.8).

Complex numbers are entered on the command line as follows:

Command: {x:y} (cartesian form)

Command: POL{r:a} (polar form)

where x is the real part and y the imaginary part or, in the polar form, r is the magnitude and a is the angle. Do not separate the { } from the numbers with spaces; consider them to be part of the number.

2.2.3 NUMBER TYPES

The ASYSTANT distinguishes six different types of numbers: integers, double precision integers, real numbers, double precision real numbers, complex numbers, and double precision complex numbers. Table 2.1 shows memory sizes, maximum array sizes, ranges, and sample input formats for each of these data types.

Data Type	Memory (Bytes)	Sample Input Formats	Maximum Number of Elements	Range
Integer	2	123S -234S	32K	Min: -32768 Max: 32767
DP Integer	4	123 -234	16K	Min: -2147483647 Max: 2147483647
Real	4	1.23 1.23E-2	16K	Min: 8.43×10^{-37} Max: 3.37×10^{38}
DP Real	8	1.23D 1.23E-2D	8K	Min: 4.19×10^{-307} Max: 1.67×10^{308}
Complex	8	{1.23:2.34} POL{1.2:3.14}	8K	(Cartesian) (Polar)
DP Complex	16	{1.23D:2.34D} POL{1.2D:3.14E2D}	4K	(Cartesian) (Polar)

TABLE R2.1 *Data Types*

2.3 Basic Arithmetic

2.3.1 BASIC OPERATIONS

The basic arithmetic operations (+, -, /, * and ^) can be executed by simply pressing the associated key. This will call up the command line with the selected operation as the first character. Pressing the **Enter** key will then cause the operation to be carried out. Additionally, these operators may be imbedded with other input on the command line. If RPN is used, the operators must be separated from other numbers or operators by a space.

The operations are applied to the top two elements of the stack in the order that they were pushed on the stack. The second entry from the top is used for the first operand, and the top stack entry is used for the second operand. The result is left on the stack in place of the two operands. Thus, for example, when the (-) operation is executed the contents of the stack are modified as follows:

Stack Before Subtraction		Stack After Subtraction
<entry 1>		<entry2>-<entry1>
<entry 2>		<entry 3>
<entry 3>	----->	<entry 4>
<entry 4>		<entry 5>
<entry 5>		

2.3.2 SYNTAX

Numbers and operations may be entered in RPN (Reverse Polish Notation) or using conventional algebraic syntax.

Reverse Polish Notation

In RPN, operator precedence is sequential from left to right and no parentheses are allowed. Each operation is applied to the top stack entry (or entries) as it is encountered on the command line. As each number (or Variable or Parameter name) is encountered it is pushed onto the top of the stack. *Operators, numbers, Variables, and Parameters must be separated from each other by a space when using RPN.*

Using RPN, the command:

Command: 6 3 / 4 +

would be performed as follows. First 6 would be pushed onto the stack, followed by 3 (leaving 3 at the top). Then the division would be performed by popping the top entry, 3, off the stack as the divisor, then popping the next entry, 6, as the dividend, and pushing the quotient, 2, back onto the stack. Next, 4 is pushed onto the stack and the addition is performed on the top two entries, 4 and 2, which are popped off the stack for the computation. Finally, the sum, 6, is pushed onto the stack. This calculation is performed so quickly that only the final result, 6, is displayed in the **Stack Contents** window.

Algebraic Syntax

Algebraic syntax may be used on the command line by pressing the "\" key as the *first character* to call up the command line window. Thereafter, the command may be entered using the conventional syntax. No spaces between entries are required, although they may be used if desired. Thus the above example could be entered as follows:

Command: \ 4+(6/3)

The result, 6, would appear in the **Stack Contents** window as before. ASYSTANT translates algebraic syntax to RPN internally and performs all computations in RPN.

Note: In algebraic syntax, arguments for User Functions must be enclosed in parentheses and must follow the function name. For example, if F1 was assigned the command SIN and you wished to take the sine of .5, then enter the command

Command: \ F1(.5)

Also, command words, such as STORE, should not be used in algebraic expressions.

2.3.3 RESULTANT DATA TYPES

The data type of the result will be complex if either operand is complex, it will be non-integer if either operand is non-integer, and it will be double precision if either operand is double precision. Please note that in the ASYSTANT *integer quotients are rounded to the nearest whole number rather than truncated.*

2.4 Arrays

2.4.1 ARRAY AND SUBARRAY DEFINITIONS

ASYSTANT arrays may be one dimensional (vectors) or two dimensional (matrices). Arrays cannot exceed 64 Kbytes in overall size. The maximum number of elements in an array can be determined by dividing 64 Kbytes by the number of bytes required for the array data type (see Table 2.1 in Section 2.2). Section 2.7 describes the various array handling options available in the Array Operations menu.

Arrays may be created via the menu options **aedit** (the array editor), **n:ramp**, and **nm:ramp**. As well, Variable arrays can be used to hold and/or transfer data files or subfiles (see Chapter 7, **File I/O**). Variable arrays also store data arrays, or waveforms, generated by ASYSTANT Main Menu options, such as **Statistics** and the **Wave Generator**.

Array elements are specified by their indices. On the command line, individual array elements may be specified by using the **[]** operators. For example, the fifth element of variable **R** is specified by typing in RPN:

Command: **R [5]**

Note: Spaces must separate each character in this line! In algebraic notation, no spaces are necessary: **\R[5]**.

Two-dimensional arrays are handled by the row and column index; separated by a comma. For example:

Command: **S [5 , 1]**

specifies row 5, column 1. Again, remember that in RPN, each character (including the comma) must be separated by a space.

Elements may also be edited with **aedit** and extracted from an array with **xsect**.

A subarray is a rectangular subset of an array. A subarray of a one-dimensional array, **V**, is specified on the command line by **V <s,n,i>**, where **s** is the index of the starting element, **n** is the count of elements to include in the subarray, and **i** is the increment of extraction. The increment **i** determines which elements are to be included in the subarray: an increment of 1 includes every element from the starting element, an increment of 2 includes every other element, and so on. A subarray of a two dimensional array, **W**, is specified by **W <r,n,i;c,m,j>**, where the first three numbers specify the row starting index, length, and increment and the last three numbers denote the column starting index, length, and increment. Subarrays must be wholly contained within their parent arrays. Subarrays may also be addressed via **aedit**, **sub**, and **xsect**.

Note: In RPN, angle brackets **< >** and punctuation must be separated from numbers by spaces.

2.4.2 ARRAY ARITHMETIC

Arithmetic can be performed on arrays as well as numbers, provided that the dimensions of the two operands are compatible. Table 2.2 lists valid combinations for the two operands, and shows the dimensions of the result in each case. If one of the operands is a number then it will be combined with each element of the other operand. If the operands are both arrays and have equal dimensions, then the operation will be applied on an element by element basis. If one operand is a two dimensional array, and the other one dimensional, then each element of the one dimensional array will be combined with each column of the two dimensional array.

Operand 1	Operand 2	Result
number	number	number
number	Dim[n] Array	Dim[n] Array
number	Dim[n,m] Array	Dim[n,m] Array
Dim[n] Array	number	Dim[n] Array
Dim[n,m] Array	number	Dim[n,m] Array
Dim[n] Array	Dim[n] Array	Dim[n] Array
Dim[n,m] Array	Dim[n,m] Array	Dim[n,m] Array
Dim[n,m] Array	Dim[m] Array	Dim[n,m] Array
Dim[m] Array	Dim[n,m] Array	Dim[n,m] Array

TABLE R2.2 *Array Arithmetic*

The arrays that are produced as results of these operations are new (unnamed) arrays. The contents of the original input arguments are not modified.

TIP

One dimensional arrays may be used as either row-or-column vectors in matrix operations. (1 x n) or (n x 1) arrays are actually two dimensional and therefore may have incompatible dimensions in operations with some arrays where a vector would not. In these cases, use **reshape** to convert the array to a vector (see section 2.7).

There are two ways to modify the contents of Variables or Parameters. The first is to use the **store** function, which assigns to the Variable or Parameter the exact contents, data type, and dimensions of the entry on the top of the stack (see section 2.6).

The other method is to use the assignment operation. This operation is performed by typing "=" on the command line and causes the contents of the second stack entry to be assigned into the Parameter or Variable entry on the top of the stack. After execution, both entries are removed from the stack and the result is stored in the target Variable or Parameter.

The = key can be used to modify values stored in Parameters or Variables, without affecting their data type or dimensions. It has a number of different applications. For instance, it can

- * fill an array, subarray, or array element with a specified value,
- * fill an array or subarray with the elements of a different array (in which case the two operands must be arrays of the same shape), or
- * fill all rows of a two dimensional array with the elements of a one dimensional array. (In this case the first operand (2nd stack entry) should be one dimensional, and the second operand (top stack entry) should be two dimensional with row length equal to the size of the first operand.)

For example,

Command: \ C=sqrt(A+B) (algebraic)
 or
 Command: A B + sqrt C = (RPN)

will store the square root of the sum of the Parameters A and B in Parameter C.

Command: \ X<2,3;1,3>=Y<1,3>+Z<2,3>

The above command will assign the sum of the Y and Z subarrays (of length 3) repeatedly into each row of the 3x3 subarray of X.

Note: In the course of making an assignment the value(s) of the first operand are converted to the data type of the second operand. This can generate overflow errors if the values being transmitted fall outside of the range that is permitted by the target data type.

2.5 Editing the Command Line

The command line may be recalled and edited. A command line buffer stores the last 256 characters entered, maintaining them as distinct command lines. When the command line window appears, sequences are "paged" backwards with F3 and forwards with F4. They may then be edited. When **Enter** is pressed, the command sequence will be executed as it appears on the command line (the cursor does not have to be at the end of the line). The following list describes the editing keys that are active on the command line and their functions.

- | | |
|--------------------|---|
| F3 | -- Recall the last command line entered. Each time F3 is subsequently pressed, the preceding command sequence is recalled from the buffer. The stored command line will overwrite any characters on the current command line from the cursor position to the right. Characters to the left of the cursor will not be affected. (Page backward). |
| F4 | -- Recall the next command sequence that was entered. The stored command line will overwrite any characters on the current command line from the cursor position to the right. Characters to the left of the cursor will not be affected. (Page forward). |
| Del | -- Delete the character at the current cursor position. |
| Ins | -- Toggle between overwrite and insert modes. |
| left arrow | -- Move the cursor to the left. |
| right arrow | -- Move the cursor to the right. |

2.6 Calculator Functions

Stack Contents					Parameters				
(empty stack)					A= .0000				
					B= .0000				
					C= .0000				
					D= .0000				
					E= .0000				
					F= .0000				
					G= .0000				
					H= .0000				
					I= .0000				
Calculator Functions					Variables				
next	store	pedit	status	print	R= .0000				
dup	drop	swap	roll	pi	S= .0000				
sin	asin	sinh	asinh	inv	T= .0000				
cos	acos	cosh	acosh	neg	U= .0000				
tan	atan	tanh	atanh	abs	V= .0000				
exp	ln	10exp	log	sqrt	W= .0000				
Main Menu Options					X= .0000				
Acquire	Graphics	Functions			Y= .0000				
File Proc	File I/O	Polys			Z= .0000				
Wave Proc	Curve Fit	Diff Eqs							
Wave Gen	Stats	Save/Exit							

SCREEN R2.1

The Calculator Functions menu contains options for stack manipulations, standard elementary functions, and several miscellaneous operations such as **store**, **print**, **pedit** and **status**. All functions act on the top entry or entries of the stack, and work with all data types including complex numbers. However, *angular units must be in radians for functions to work properly on complex arguments* (see **status** below). When a function is applied to an array, the function is applied to each element of the array. The result is a new unnamed array of the same shape as the original.

next

Advances to the next menu: Array Operations

store

Stores the entry on the top of stack in a selected Parameter or Variable. When this option is exercised, a prompt appears in the bottom left corner of the display for the letter of a Parameter or Variable. The top stack entry is popped off the stack into the selected Parameter or Variable.

Note: Arrays cannot be stored in Parameters, only in Variables.

pedit

The **Parameter editor** is used to edit the values stored in Parameters. A window is opened in which a list of the Parameters and their values is displayed for editing. The **Esc** key accepts the edited values and returns to the Desk Calculator.

Note: The data type of a Parameter cannot be modified by this option, only its value can be changed.

status

Allows for modification of the current system status. A window is opened in which a list of the current status values is displayed for editing. The **Esc** key accepts the edited values and returns to the Desk Calculator.

There are three aspects to the system status: numeric output format, angular units, and data type.

Numeric Output Format

The output format determines the way that the ASYSTANT displays and prints numbers. Numeric output can be in exponential or decimal format with a specified number of decimal places, and can be of fixed or variable length. The size of the field used for fixed length formats should include the digits, sign, decimal point, and 5 characters for the exponent if the format is exponential. The output format for complex numbers can be toggled between cartesian (real and imaginary) and polar (magnitude and angle).

Angular units

The angular units can be either degrees, radians, or gradians. These units are used by in trigonometric functions and in the conversion operations described in section 2.8.

Data Type

The system data type can be integer, double precision integer, real, double precision real, complex, or double precision complex. The data type is referenced by the **n:ramp** and **nm:ramp** array operations to determine the number type of the generated array (see section 2.7).

print

Outputs the value(s) of the entry on the top of stack to the display, printer, or the note file (disk). If disk output is selected, a second prompt allows output to be inserted at the current cursor location. Otherwise, the values are appended to the end of the note file. Output to each device can be paused and restarted by pressing the space bar. Output can be terminated by pressing the space bar three times in rapid succession. Print uses the numeric output format specified in the status option.

dup

Duplicates the entry on the top of the stack.

drop

Drops the entry on the top of the stack.

swap

Swaps the first and second stack entries.

roll

Rolls the contents of the stack so that the last entry is brought to the top, and the other entries are pushed down by one.

pi

Pushes the double precision value of π on the top of stack.

sin

Computes the sine of the top stack entry.

asin

Computes the arcsine of the top stack entry.

sinh

Computes the hyperbolic sine of the top stack entry.

asinh

Computes the hyperbolic arcsine of the entry on the top of stack.

inv

Takes the reciprocal of the entry on the top of stack.

cos	Computes the cosine of the entry on the top of stack.
acos	Computes the arccosine of the entry on the top of stack.
cosh	Computes the hyperbolic cosine of the entry on the top of stack.
acosh	Computes the hyperbolic arccosine of the entry on the top of stack.
neg	Takes the additive inverse of the number on the top of stack.
tan	Computes the tangent of the entry on the top of stack.
atan	Computes the arctangent of the entry on the top of stack.
tanh	Computes the hyperbolic tangent of the entry on the top of stack.
atanh	Computes the hyperbolic arctangent of the entry on the top of stack.
abs	Takes the absolute value of the entry on the top of stack.
exp	Computes the base e exponential of the entry on the top of stack.

ln

Computes the natural log of the entry on the top of stack.

10exp

Computes the base 10 exponential of the entry on the top of stack.

sqrt

Takes the square root of the entry on the top of stack.

2.7 The Array Operations Menu

Stack Contents				Parameters	
(empty stack)				A=	.0000
				B=	.0000
				C=	.0000
				D=	.0000
				E=	.0000
				F=	.0000
				G=	.0000
				H=	.0000
				I=	.0000
Array Operations				Variables	
next	xsect	sub	trans	R=	.0000
aedit	diag	n:ramp	nm:ramp	S=	.0000
reverse	n:rot	cat	lam	T=	.0000
[]sum	[]prod	[]max	[]min	U=	.0000
^sum	^max	^min	reshape	V=	.0000
sort	index	lookup	n:search	W=	.0000
Main Menu Options				X=	.0000
Acquire	Graphics	Functions		Y=	.0000
File Proc	File I/O	Polys		Z=	.0000
Wave Proc	Curve Fit	Diff Eqs			
Wave Gen	Stats	Save/Exit			

SCREEN R2.2

This menu provides operations for creating and manipulating arrays (see section 2.4 for array definitions). The array functions can be grouped as follows:

Array Generation: `n:ramp`, `nm:ramp`, and `aedit` are options for generating new unnamed arrays, and for editing array elements.

Array Access: `xsect`, `sub`, `trans`, `diag`, `reverse` are options that alter the way that elements of a parent array are accessed, without actually generating a new array. Thus, for example, when a number is assigned into a subarray (using `=`) the contents of the parent array will also be changed.

Array Reordering: `n:rot`, `reshape`, `sort`, `lookup` are operations that generate new arrays that contain the same values as the input array, only in a different sequence.

Array Indexing: `index` and `n:search` generate an array of indices that correspond to a reordering or a specialized subset of the input array.

Array Catenation: `lam` and `cat` are used to combine two arrays together to form a single large array.

Cumulative Operations: []sum, []prod, []max, and []min carry out cumulative operations along rows.

Running Cumulative operations: ^sum, ^max, and ^min carry out running cumulative operations along rows.

The menu options are described individually below.

next

Advances to the next menu: Conversions and Special Functions.

xsect

Extracts a column, row or individual element from the array on the top of the stack. Prompts appear requesting a column number and, if the array is two dimensional, a row number. To extract an entire column, select the column number and leave the row number equal to -1. Similarly, to extract an entire row, the column number should be -1. To extract an individual element, specify both the column and row indices. The Esc key executes the extraction.

sub

Extracts a subarray from the array on the top of the stack. Prompts appear requesting a column start index, length, and increment. If the array is two dimensional then additional prompts request a starting row index, length, and increment. The start indices specify the first array element in the subarray and the length specifies the number of elements to extract. The increment determines which elements are extracted: an increment of 1 will extract every element from the starting element, an increment of 2 will extract every other element, and so on. Each time a start index or increment value is modified, the corresponding size value is checked to see if it falls within the limits of the parent array. If it doesn't fit, then it is automatically reduced to the largest allowable value. The Esc key executes the subarray extraction

trans

Transposes the two dimensional array on the top of stack.

aedit

The **array editor** creates and/or displays a Variable array for editing. A prompt appears requesting a Variable name. If the selected Variable already contains an array, it is displayed on the screen ready for editing. Otherwise, an array will be created and stored in the named Variable. In the latter case, prompts request the dimensions and data type to use in generating the array. The new array is then displayed on the screen, filled with zeroes, and ready for editing.

The editor displays the array in a row/column format that is controlled similarly to a standard electronic spreadsheet. A highlight denotes the current element and may be scrolled through the array with the use of the **arrow** keys. The value of the current element is changed by entering the new value in the editor's command line at the bottom of the editing screen. Numbers are input in the same format as on the command line (see Table 2.1 in section 2.2). Each time a number is entered, the current element highlight moves one step in the direction of the most recently pressed **arrow** key. Array rows and columns may be copied, inserted, or deleted. Array editing is terminated by pressing **Enter** alone (without typing a number). The following describes key usage in the array editor.

Home	-- Moves the highlight to the first array element.
End	-- Moves the highlight to the first element of the last row.
PgUp	-- Moves the highlight to the last element of the first row.
PgDn	-- Moves the highlight to the last element of the last row.
F7	-- Inserts a new column of 0's to the right of the current column.
F8	-- Inserts a new row of 0's just below the current row. This operation is not valid on one dimensional arrays.
Shft-F7	-- Deletes the current column.
Shft-F8	-- Deletes the current row. This operation is not valid on one dimensional arrays.
F9	-- Marks the current column as the source for a column to column copy.
F10	-- Marks the current row as the source for a row to row copy. This operation is not valid on one dimensional arrays.
Shft-F9	-- Copies the column marked by F9 into the current column. The previous column values are overwritten.
Shft-F10	-- Copies the row marked by F10 into the current row. The previous row values are overwritten. This operation is not valid on one dimensional arrays.

diag

Extracts the main diagonal of a two dimensional array.

n:ramp

Generates a one dimensional ramped array of length equal to the number on the top of the stack. The input argument *n* is removed from the stack and replaced by the array. In a ramped array, the value of the *i*-th array element is assigned the value *i*. The array elements are of the current data type, as set in **status** (see section 2.6).

nm:ramp

Generates a two dimensional ramped array. The row dimension is taken from the second stack entry and the column dimension from the top stack entry. These numbers are removed from the stack and are replaced by the array. The *ij*-th array element is assigned the value $(i-1)m+j$, where *m* is the column dimension. The array elements are of the current data type, as set in **status** (see section 2.6).

reverse

Changes the way that the elements of the array on the top of stack are accessed, by indexing the columns in reverse order.

n:rot

Generates an array with the same values as the input array, but with the columns rotated to the right by *n* places. The input array is passed to **n:rot** as the second stack entry, and the shift factor *n* is passed as the top stack entry. The rotation is cyclic, so that elements in the rightmost column are rotated into the leftmost column. A negative value for the shift factor will cause the array values to be rotated to the left.

cat

Catenates the two arrays on the top of stack to form a single large array. The input arguments must have the same number of rows. In the resultant array, the columns of the input array from the top of stack will follow the columns of the input array that was second on the stack.

lam

Laminates the two arrays on the top of stack to form a single large array. The input arguments must have the same number of columns. In the resultant array, the rows of the input array from the top of stack will follow the rows of the input array that was second on the stack.

[]sum

Computes the cumulative sum of each row of the array on the top of the stack. The resultant one-dimensional array or number replaces the original array on the top of the stack.

[]prod

Computes the cumulative product of each row of the array on the top of the stack. The resultant one dimensional array or number replaces the original array on the top of the stack.

[]max

Computes the cumulative maximum of each row of the array on the top of the stack. The resultant one dimensional array or number replaces the original array on the top of the stack.

[]min

Computes the cumulative minimum of each row of the array on the top of the stack. The resultant one dimensional array or number replaces the original array on the top of the stack.

^sum

Computes the cumulative running sum of each row of the array on the top of stack. The resultant array of sums is of the same dimension as the original array and replaces it on the top of the stack.

^max

Computes the cumulative running maximum of each row of the array on the top of stack. The resultant array is of the same dimension as the original array and replaces it on the top of the stack.

^min

Computes the cumulative running minimum of each row of the array on the top of stack. The resultant array is of the same dimension as the original array and replaces it on the top of the stack.

reshape

Generates an array that contains the same values as the input array, but has a different shape. Prompts appear requesting the row and column dimensions of the output array. Whenever one of these two values is modified, the other is automatically adjusted so that the overall size of the output array will be as close as possible but not exceeding the size of the input array. The elements are taken from the input array row by row, and are entered in the output array following the same sequence. The Esc key executes the reshape operation.

sort

Sorts the array on the top of stack so that its elements occur in ascending order by value. The resulting array is one dimensional regardless of the shape of the input array. Complex arrays cannot be sorted.

index

Generates an array containing a list of all the indices of the array on the top of stack, organized by ascending value order of the elements. Index cannot be used on complex arrays.

lookup

Uses the array on the top of stack as an index list to sort the values in the array that is the second entry on the stack. (Note: the command sequence **dup index lookup** is equivalent to **sort**).

n:search

Searches through an array to find all elements that satisfy a selected inequality, and then returns a list containing the indices of those elements. The following menu of options is offered:

```
=n:search  
<n:search  
>n:search  
<>n:search  
return
```

The input array should be the second stack entry, and the search parameter **n** should be on the top of stack. The search is carried out as soon as the inequality is selected.

2.8 Conversions and Special Functions

Stack Contents				Parameters	
(empty stack)				A=	.0000
				B=	.0000
				C=	.0000
				D=	.0000
				E=	.0000
				F=	.0000
				G=	.0000
				H=	.0000
				I=	.0000
Conversions and Special Functions				Variables	
next	conj	mod	labels	R=	.0000
xy>z	z>xy	ra>z	z>ra	S=	.0000
xy>pol	pol>xy	xyz>sph	sph>xyz	T=	.0000
erf	n!	n:C:r	n:P:r	U=	.0000
Jn	Yn	sJn	sYn	V=	.0000
K(m)	E(m)	Gamma	Betax(a,b)	W=	.0000
Main Menu Options				X=	.0000
Acquire	Graphics	Functions		Y=	.0000
File Proc	File I/O	Polys		Z=	.0000
Wave Proc	Curve Fit	Diff Eqs			
Wave Gen	Stats	Save/Exit			

SCREEN R2.3

This menu has options for conversions between real and complex numbers, polar and cartesian coordinates, and spherical and cartesian coordinates. Conversions may be performed on scalars or arrays. Also available on this menu are a number of special functions for advanced operations.

next

Advances to the next menu: Wave and Matrix Operations.

conj

Takes the complex conjugate of the entry on the top of stack.

mod

Computes the modulo of the second stack entry relative to entry on the top of stack.
(i.e. <2nd entry> mod <1st entry>). Modulo is defined by the formula

$$a \bmod b = \text{sgn}(b) * (a - [a/b] * b)$$

where $[]$ denotes the greatest integer function.

labels

Assigns labels to Parameters and Variables. Each label may be up to 7 characters in length and all ASCII characters are valid. The labels are displayed in an editable list that is exited by pressing the **Esc** key. Edited labels are displayed beside the Parameters and Variables letters in the Desk Calculator windows.

xy>z

Converts the top two stack entries into a single complex entry by using the second entry for the real part and the top entry as the imaginary part.

z>xy

Splits the complex entry on the top of stack into its real and imaginary parts. The imaginary part is left on the top of stack and the real part is pushed beneath it.

ra>z

Converts the top two stack entries into a single complex entry by using the second entry for the magnitude and the top entry as the angle. The **status** should be set to reflect the angular units used before the conversion is performed (see section 2.6, **status**).

z>ra

Splits the complex entry on the top of stack into polar components. The angles are left on the top of stack and the magnitude pushed underneath. The angles are reported in the current angular units (see section 2.6, **status**).

xy>pol

The two entries on the top of stack are converted from cartesian to polar coordinates. The y values are taken from the top of the stack and the x values from underneath. The angles are returned on the top of stack with the magnitudes underneath. The angles are reported in the current angular units (see section 2.6, **status**).

pol>xy

The two entries on the top of stack are converted from polar to cartesian coordinates. Angles are taken from the top of the stack and the magnitudes from underneath. The y values are returned on the top of the stack with the x values underneath. The **status** should be set to reflect the angular units used before the conversion is performed (see section 2.6, **status**).

xyz>sph

The three entries on the top of stack are converted from cartesian to spherical coordinates. The z values are taken from the top stack entry, the y values from the second entry, and the x values from the third entry. The Azimuthal angles are returned on the top of the stack with the polar angles and magnitudes as the second and third stack entries, respectively. The angles are reported in the current angular units (see section 2.6, **status**).

sph>xyz

The three entries on the top of stack are converted from spherical to cartesian coordinates. The Azimuthal angles are taken from the top stack entry with the polar angles and magnitudes underneath. z values are returned on the top of stack with the y and x values underneath. The **status** should be set to reflect the angular units used before the conversion is performed (see section 2.6, **status**).

erf

Computes the error function of the top stack entry. The error function is defined by the formula

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

n!

Computes the factorial of the top stack entry.

n:C:r

Computes the number of ways to choose r objects from a collection of n objects, not distinguishing between permutations. The r values are taken from the top of stack and the n values are taken from the second stack entry.

n:P:r

Computes the number of ways to choose r objects from a collection of n objects, where different permutations are counted as distinct choices. The r values are taken from the top of stack and the n values are taken from the second stack entry.

Jn

Computes the Bessel function $J_n(x)$ of the entry on the top of stack. The order n is taken from the second stack entry. $J_n(x)$ is defined by the formula

$$J_n(z) = \frac{1}{\pi} \int_0^\pi \cos(z \sin u - nu) du$$

Yn

Computes the Bessel function $Y_n(x)$ of the entry on the top of stack. The order n is taken from the second stack entry. $Y_n(x)$ is defined by the formula

$$Y_n(z) = \lim_{r \rightarrow n} \frac{J_n(z) \cos(r\pi) - J_{-n}(z)}{\sin(r\pi)}$$

sJn

Computes the spherical Bessel function $sJ_n(x)$ of the entry on the top of stack. The order n is taken from the second stack entry. $sJ_n(x)$ is defined by the formula

$$sJ_n(z) = \sqrt{\frac{\pi}{2z}} J_{\frac{2n+1}{2}}(z)$$

sYn

Computes the spherical Bessel function $sY_n(x)$ of the entry on the top of stack. The order n is taken from the second stack entry. $sY_n(x)$ is defined by the formula

$$sY_n(z) = \sqrt{\frac{\pi}{2z}} Y_{\frac{2n+1}{2}}(z)$$

K(m)

Computes the elliptic integral $K(m)$ of the entry on the top of stack. The order m is taken from the second stack entry. $K(m)$ is defined by the formula

$$K(m) = \int_0^{\pi/2} [1 - m \sin^2 u]^{-1/2} du$$

E(m)

Computes the elliptic integral $E(m)$ of the entry on the top of stack. The order m is taken from the second stack entry. $E(m)$ is defined by the formula

$$E(m) = \int_0^{\pi/2} [1 - m \sin^2 u]^{1/2} du$$

Gamma

Computes the gamma function of the entry on the top of stack. The gamma function is defined by the formula

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$$

Betax(a,b)

Computes the incomplete beta function $Beta[x;a,b]$ of the third entry on the stack. The a and b parameters are taken respectively from the second and first stack entries. $Beta[x;a,b]$ is defined by the formula

$$B_x(a,b) = \int_0^x t^{a-1} (1-t)^{b-1} dt$$

2.9 Wave and Matrix Operations

Stack Contents				Parameters	
{empty stack}				A=	.0000
				B=	.0000
				C=	.0000
				D=	.0000
				E=	.0000
				F=	.0000
				G=	.0000
				H=	.0000
				I=	.0000
Wave and Matrix Operations				Variables	
next	smooth	integ	diff	R=	.0000
fft	ifft	2d:fft	i2d:fft	S=	.0000
pwrspec	autocorr	convolve	window	T=	.0000
Hilbt	crosscorr	avg	stdev	U=	.0000
max	min	solve:eqs	trace	V=	.0000
matrix*	kronec*	determ	inv\mat	W=	.0000
Main Menu Options				X=	.0000
Acquire	Graphics	Functions		Y=	.0000
File Proc	File I/O	Polys		Z=	.0000
Wave Proc	Curve Fit	Diff Eqs			
Wave Gen	Stats	Save/Exit			

SCREEN R2.4

This menu contains a set of functions that are used to manipulate and analyze waveforms. It also contains options for carrying out common matrical operations.

In the ASYSTANT, a single waveform is nothing more than a one dimensional array, and a series of waveforms is represented by the successive rows of a two dimensional array. Vectors and matrices are also represented by one and two dimensional arrays; so both waveform and matrix arithmetic can be performed as basic array operations.

next

Advances to the next menu: Calculator Functions.

smooth

Smooths the array on the top of stack using a Blackman window low pass filter with a specified cutoff frequency. If the array is two dimensional then each row will be smoothed. A prompt is displayed requesting the cutoff frequency in units of cycles/pt. The frequency response of the filter tapers off to 0 at the cutoff frequency. The cutoff frequency is limited to the range (.03-.5), which corresponds to wave components with periods ranging

between 33 and 2 points. The cutoff frequency is also limited (on the .03 side) by the size of the waveform being smoothed. A rule of thumb is that the cutoff frequency should not be less than $2/n$, where n is the number of points of the input waveform.

integ

Computes the running integral of the array on the top of the stack using the Simpson's 1/3 rule. If the array is two-dimensional, then each row will be integrated.

diff

Computes a derivative of specified order (up to 7) of the array on the top of the stack. If the array is two-dimensional then the differentiation will be carried out on each row. Interpolating polynomials of specified degree (up to 7) are used to calculate the derivative. Prompts appear requesting the derivative order and polynomial degree. The order of the derivative cannot exceed the polynomial degree.

fft

Computes the fast Fourier transformation of the array on the top of the stack. The algorithm requires that the data is a power of two in length (max=4096), so before the fft is executed the input waveform is truncated to the nearest power of 2. If the input array is two dimensional then each row will be transformed. The Fourier transformation is defined by the formula

$$\hat{f}(k) = \sum_{j=0}^{n-1} f(j) e^{-2\pi i \frac{jk}{n}} \quad (k = 0, 1, \dots, n-1)$$

ifft

Computes the inverse Fourier transformation of the array on the top of the stack. The algorithm requires that the data is a power of two in length (max=4096), so before the fft is executed the input waveform is truncated to the nearest power of 2. If the input array is two dimensional then each row will be transformed. The inverse Fourier transformation is defined by the formula

$$f(j) = \frac{1}{n} \sum_{k=0}^{n-1} \hat{f}(k) e^{+2\pi i \frac{jk}{n}} \quad (j = 0, 1, \dots, n-1)$$

2d:fft

Computes the two dimensional fast Fourier transformation of the two dimensional array on the top of the stack. The algorithm requires that both dimension are a power of two in length (max=32x32), so before the fft is executed the input array is truncated in both dimensions to the nearest power of 2. The transformation is defined by the formula

$$\hat{f}(k_1, k_2) = \sum_{j_2=0}^{n_2-1} \sum_{j_1=0}^{n_1-1} f(j_1, j_2) e^{-2\pi i \left(\frac{j_1 k_1}{n_1} + \frac{j_2 k_2}{n_2} \right)} \quad (k_1 = 0, 1, \dots, n_1-1)$$

$$(k_2 = 0, 1, \dots, n_2-1)$$

2d:ifft

Computes the two dimensional inverse Fourier transformation of the two dimensional array on the top of the stack. The algorithm requires that both dimension are a power of two in length (max=32x32), so before the fft is executed the input array is truncated in both dimensions to the nearest power of 2. The transformation is defined by the formula

$$f(j_1, j_2) = \frac{1}{n_1 n_2} \sum_{k_2=0}^{n_2-1} \sum_{k_1=0}^{n_1-1} \hat{f}(k_1, k_2) e^{+2\pi i \left(\frac{j_1 k_1}{n_1} + \frac{j_2 k_2}{n_2} \right)} \quad (j_1 = 0, 1, \dots, n_1 - 1) \\ (j_2 = 0, 1, \dots, n_2 - 1)$$

pwrspec

Computes the power spectrum of the array on the top of the stack. The algorithm requires that the data is a power of two in length (max=4096), so before the Power Spectrum is executed the input waveform is truncated to the nearest power of 2. If the input array is two dimensional then the power spectrum of each row will be calculated. The power spectrum is defined to be the squared magnitude of the Fourier transformation.

TIP

For real signals, both the fft and power spectrum return a symmetrical waveform showing both real frequencies and aliased frequencies. It is conventional to display the real frequencies only and to scale the X axis for frequency in Hz. This requires two simple manipulations on the raw output of the fft or pwrspec functions, which can easily be stored in a function key. You may wish to store these definitions in a setup file if you use these operations frequently.

We assume that the power spectrum or fft is on top of the stack, and that you have stored the sampling rate used to capture the original signal (units of Hz) in Parameter A, and the number of points in the fft in Parameter B. (Remember, if the number of data points in your original data scan does not equal a power of 2, the fft or power spectrum will be truncated.) The following user function definition will store a frequency index array in X and the fft or power spectrum in Y. Each will be truncated to show real frequencies only:

```
< 1 , B 2 / 1 + >
STORE Y
B N:RAMP 1 - B / A *
< 1 , B 2 / 1 + >
STORE X
```

Remember to separate each operator with a space! To plot the data, you can select select **Graphics** and perform an xy Auto or xy Plot.

autocorr

Computes the autocorrelation function of the waveform on the top of the stack. If the input array is two dimensional then the autocorrelation will be calculated for each row.

convolve

Computes the aperiodic convolution of the first and second stack entries. If both input arrays are two dimensional then they must have the same number of rows. In that case the convolution is computed row by row. If one input array is two dimensional and the other is one dimensional, then the one dimensional array is convolved with each column of the two dimensional array.

window

Applies a Blackman window to a specified subset of the waveform on the top of stack. If the top stack entry is two dimensional then the window will be applied to each row. Prompts request an index for the starting and ending data points for the window. All data elements outside of the window will be set equal to 0. The Blackman window for the normalized range $[-1,1]$ is defined by the formula

$$.42 + .5 \cos(\pi x) + .08 \cos(2\pi x)$$

Hilbt

Computes the Hilbert transform of the array on the top of stack. The algorithm requires that the data is a power of two in length (max=4096), so before the transform is executed the input waveform is truncated to the nearest power of 2. If the input array is two dimensional then each row will be transformed. The Hilbert transform is defined by the formulae

$$\hat{h}(k) = \begin{cases} i \hat{f}(k) & k = 1, 2, \dots, n/2 - 1 \\ 0 & k = 0 \\ -i \hat{f}(k) & k = n/2 + 1, \dots, n - 1 \end{cases}$$

$$\hat{f}(k) = \sum_{j=0}^{n-1} f(j) e^{-2\pi i \frac{jk}{n}} \quad (k = 0, 1, \dots, n-1)$$

crosscorr

Computes the cross correlation of the first and second stack entries. If both input arrays are two dimensional then they must have the same number In that case the correlation is computed row by row. If one input array is two dimensional and the other is one dimensional, then the one dimensional array is correlated with each column of the two dimensional array.

avg

Computes the average value of a waveform, or an average waveform of a series of waveforms. If the array on the top of stack is one dimensional then **avg** will compute its average value. If the array is two dimensional then an average waveform will be computed by averaging down columns.

stdev

Computes the sample standard deviation of a waveform, or a standard deviation waveform of a series of waveforms. If the array on the top of stack is one dimensional then **stdev** will compute the sample standard deviation of the data values. If the array is two dimensional then a standard deviation waveform will be computed by taking the sample standard deviation down columns. The sample standard deviations is defined by the formula

$$\sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2}$$

max

Computes the maximum of the two entries on the top of the stack. If one or both of the entries are arrays then the maximum is computed on an element by element basis following the same conventions that apply to the basic arithmetic operations (see section 2.3)

min

Computes the minimum of the two entries on the top of the stack. If one or both of the entries are arrays then the minimum is computed on an element by element basis following the same conventions that apply to the basic arithmetic operations (see section 2.3)

solve:eqs

Finds the solution vector **x** to the matrix equation $y = Ax$, where **A** is an $n \times n$ matrix and **y** is an n vector. The **y** vector is input on the top of the stack with the **A** matrix underneath. The solution vector **x** will replace the two input arguments on the top of stack.

trace

Computes the trace of the matrix on the top of the stack and replaces it with the result.

matrix*

Computes the matrix product of two arrays. The array on the top of stack is multiplied by the array just beneath it, and the result is left on the stack in their place. Note that if the first operand (second entry on the stack) is one dimensional then it is interpreted as a $1 \times n$ matrix, and if the second operand (top entry) is one dimensional then it is interpreted as an $n \times 1$ matrix. When both operands are one dimensional then their matrix product will coincide with their inner or dot product.

kronec*

Computes the Kronecker product of two arrays. The array on the top of stack is multiplied by the array just beneath it, and the result is left on the stack in their place. Note that if the first operand (second entry on the stack) is one dimensional then it is interpreted as a $1 \times n$ matrix, and if the second operand (top entry) is one dimensional then it is interpreted as an $n \times 1$ matrix.

determ

Computes the determinant of the matrix on the top of stack and replaces it with the result.

inv\mat

Computes the inverse of the matrix on the top of the stack and replaces it with the result.

2.10 The Command Line: an Alternative to Menus

Operations from the four calculator menus can be executed at the command line by typing their names rather than selecting them as menu options. Most menu options can be executed as commands in this way, but there are a few that cannot. As well, some operations must be carried out using a different syntax. The reason for these exceptions is inherent in the use of the command line. Command lines can be executed at times when it is awkward to open additional windows (eg., in the Mini Calculator -- see Chapter 1). Therefore, menu options that open windows to prompt for additional information are not available in the command mode. In some cases, additional commands have been provided to match the capabilities of omitted prompts.

Below is a list of menu items that are either unavailable in the command mode or that carry certain requisites when used in the command mode.

A second list describes special "hidden" commands that are only available in the command mode and are not found on the calculator menus.

An example is given for each new command word.

2.10.1 MENU OPTIONS ON THE COMMAND LINE

adit

Equivalent command is not available.

diff

This command performs the same operation as the **diff** menu option, but it does not prompt for derivative order and polynomial degree. These parameters should be set with the **set.order** and **set.degree** commands (see Hidden Commands below) before **diff** is executed. The derivative is taken on the array on the top of the stack.

Command: 3 SET.ORDER 2 SET.DEGREE R DIFF

labels

Equivalent command is not available.

n:search

-- **=n:search**, **<n:search**, **>n:search**, **<>n:search** These words directly execute the options of the **n:search** menu.

Command: R .5 <>N:SEARCH

pedit

Equivalent command is not available.

print

Equivalent command is not available.

reshape -- n:reshape, nm:reshape

These commands reshape an array according to the dimensions passed on the top of the stack. **n:reshape** reshapes the array that is second on the stack into a one dimensional array, with length equal to the number on the top of stack. **nm:reshape** reshapes the third stack entry into a two dimensional array, with row and column dimensions taken from the second and first stack entries, respectively.

Command: R 10 2 NM:RESHAPE

smooth

This command performs the same operation as the **smooth** menu option, but it does not prompt for cutoff frequency. The cutoff frequency should be set with **set.frequency** command (see Hidden Commands below) before smooth is performed. The array on the top of the stack is smoothed.

Command: .1 SET.FREQUENCY R SMOOTH

status -- deg, grad, rad

These words reset the angular unit currently in use. The new units only remain in effect until the end of the command line. (When used in function definitions, these words remain in effect until the end of the function).

Command: DEG

store

store can also be executed as a command, but the target Variable or Parameter must immediately follow it on the command line.

Command: STORE R

sub

-- <row-start, row-size, row-incr; col-start, col-size, col-incr >

Subarrays can be extracted from a parent array on the top of the stack by specifying the subarray parameters in the format of the number list shown above. The punctuation (<> ,;) must be separated from the parameters by spaces if using RPN. The row parameters can be omitted if all rows are to be extracted, and they must be omitted if the parent array is one dimensional (eg., <3,12>). If only two values are specified for either the row or column dimensions (eg., <1,2;2,4>), then they are assumed to be the start and size parameters and the increment is assumed to be 1. If only 1 value is specified then it is taken to be the start value, and the size is assumed to be the largest value that will fit inside the parent array.

Command: R < 1 , 2 , 3 ; 2 , 4 , 3 >

window

-- B-window

This command will carry out the same windowing operation as window, but will not prompt for the starting and ending points of the window. These parameters should be set with the **start.window** and **end.window** commands (see Hidden Commands below) before executing **B-window**. The Blackman window is applied to the array on top of the stack.

Command: 1 START.WINDOW 10 END.WINDOW R B-WINDOW

xsect

-- [row# , col#]

Columns, rows, or individual elements can be extracted from a parent array on the top of the stack by specifying the row and column numbers in brackets separated by a comma. The brackets and comma must be separated from the parameters by spaces if using RPN. The row# can be omitted from the list if an entire column is to be extracted, and the col# should be set equal to -1 in order to extract an entire row. If the parent array is one dimensional then the row# must be omitted.

Command: R [10 , -1]

2.10.2 HIDDEN COMMANDS

bye

Exits from the ASYSTANT and returns to DOS.

clear

Clears all entries from the stack (and frees the memory that had been allocated by the stack entries).

clear.vars

Clears (sets to zero) all Calculator Variables (and frees the memory that had been allocated by the Variable contents).

end.window

Sets the index for the starting data point for the **window** operation to the value on the top of the stack. The index cannot exceed the number of points on the waveform.

mem

Copies the contents of the Waveform memory onto the top of the stack. This word can only be used when the Mini Calculator is invoked from the **Wave Gen** or **Wave Proc** menus.

>mem

Copies the contents of the top of stack into the Waveform memory. The stack entry must have dimensions compatible with the waveform that is currently in memory. This word can only be used when the Mini Calculator is invoked from the **Wave Gen** or **Wave Proc** menus.

reset

Resets (clears) the following user storage areas: the stack, the Calculator Parameters and Variables, and the user defined Polynomials. The Parameters and Variables are set to zero. The User Functions will *not* be affected by **reset**. **Reset** frees the memory that had held the storage contents. The maximum amount of memory is then available for memory consuming tasks, such as the processing of large data arrays. **Reset** is also useful for "wiping the slate clean" during an ASYSTANT session.

Note: Arrays contained in the stack and the Variables are the notable consumers of memory. Use the **clear** and **clear.vars** commands to free this memory without resetting the Parameters and Polynomials, which occupy relatively little space.

set.degree

Sets the polynomial degree for the **diff** operation to the value on the top of the stack. 7 is the maximum order allowed. The degree must be equal or less than the derivative order.

set.frequency

Sets the cutoff frequency for the **smooth** operation to the value on the top of the stack.

set.order

Sets the order of the derivative for the **diff** operation to the top of the stack. 7 is the maximum order allowed. The derivative order cannot exceed the polynomial degree (see **set.degree**).

start.window

Sets the index for the starting data point for the **window** operation to the value of the top of the stack. The index cannot exceed the number of points in the waveform.

The user function and polynomials are executed in command mode by entering the characters **Fn** or **Pn**, where n is a number 1-10.

CHAPTER 3: USER FUNCTIONS

Function Definitions				Parameters	
Press Esc when done					
F1	1) P1			COST	A= 2500
	2) 100 N:RAMP SIN STORE R			UNITS	B= 13
	3) A R * STORE S				C= .0000
	4) \ (A*2)-(12*B)				D= .0000
	5) F2 STORE T				E= .0000
Use PgUp and PgDn to edit other functions					F= .0000
Array Operations					G= .0000
	next	xsect	sub		H= .0000
	aedit	diag	n:ramp		I= .0000
	reverse	n:rot	cat		
	[]sum	[]prod	[]max		
	^sum	^max	^min		
	sort	index	lookup		
			trans		
			nm:ramp		
			lam		
			[]min		
			reshape		
			n:search		
Main Menu Options				Variables	
Acquire	Graphics	Functions		INPUT	R= 100 REAL
File Proc	File I/O	Polys			S= 100 REAL
Wave Proc	Curve Fit	Diff Eqs			T= .0175
Wave Gen	Stats	Save/Exit			U= .0000
					V= .0000
					W= .0000
					X= .0000
					Y= .0000
					Z= .0000

3.1 Overview

3.1.1 GENERAL DESCRIPTION

A sequence of commands from the desk calculator can be associated with each of the function keys F1-F10. These "user functions" can then be executed simply by pressing the appropriate function key, or by typing the characters F_n at the command line, where n is the function key number 1-10. The function keys are only active in the Desk Calculator, but as characters on the command line, the user functions can also be executed in the Mini Calculator. Consequently, user functions may be executed from within Main Menu options, such as the File Processor.

The Functions option in the Main Menu is used to define and edit the command sequences that are associated with each function key. Five command lines are available for each function. For more information on the command

syntax see Chapter 2, section 2.7. After a function is defined or edited it is compiled. If an error is encountered during the compilation, an error message appears and a null operation is assigned to the function. The command sequences can then be edited to correct the error.

3.1.2 ARGUMENTS

User functions will take arguments just as command line sequences do: arguments will be popped off the stack as needed or they may be specified in the function by Variable or Parameter names (effectively the same operation since the Variable or Parameter is automatically pushed on the stack when encountered in the command line). The contents of the Variables and Parameters at the time the function is executed will be used.

When a user function is included in a command sequence using algebraic syntax, its arguments must be enclosed in parentheses, separated from each other by commas, and should immediately follow the function name. The arguments should appear in the order in which they should be pushed on the stack. For example, if F1 was assigned the sequence - SIN, it could be executed at the command line by

Command: \ F1 (A,4)

which will subtract 4 from the contents of Parameter A and then take the sine of the result, leaving it on the top of the stack.

In RPN, the arguments should precede the function so that they are pushed onto the stack in the proper order. The same function would be specified in RPN as

Command: A 4 F1

3.1.3 NESTING

User functions may be nested, i.e., one user defined function key may be imbedded in the command sequence assigned to another function key. The number of nesting levels allowed is dependent on the availability of internal stack space. There is less internal stack space available from within the Main Menu options than from within the Desk Calculator. As a general rule of thumb, do not nest user functions more than 5 levels deep if they will be executed from the Desk Calculator and not more than 2 levels deep if executed from the Mini Calculator within a Main Menu option, eg., Graphics.

WARNING

Be careful not to nest user functions so that two functions call each other, creating an infinite loop. That is, do not include F1 in the definition of F2 if F1 calls F2.

3.2 Function Editing

When the **Functions** menu option is selected, a window is opened to display the listing of the command lines that are currently assigned to the F1 function. A flashing cursor is positioned at the first character of the first line and text can be directly entered starting at that point. The arrow keys will reposition the cursor without altering the text. The PgUp and PgDn keys page through the listings for each function F1-F10. These and other keys are used for function editing as follows:

Ins

The Insert key is used to toggle between insert and overwrite modes.

Del

The Delete key erases the character at the cursor position.

Backspace

Erases the character immediately preceding the cursor position.

Home

Positions the cursor at the first character of the first line.

End

Positions the cursor at the first character of the last line.

PgUp

Switches the display to the next function. If the current function has been modified then it forces a recompilation of the command lines.

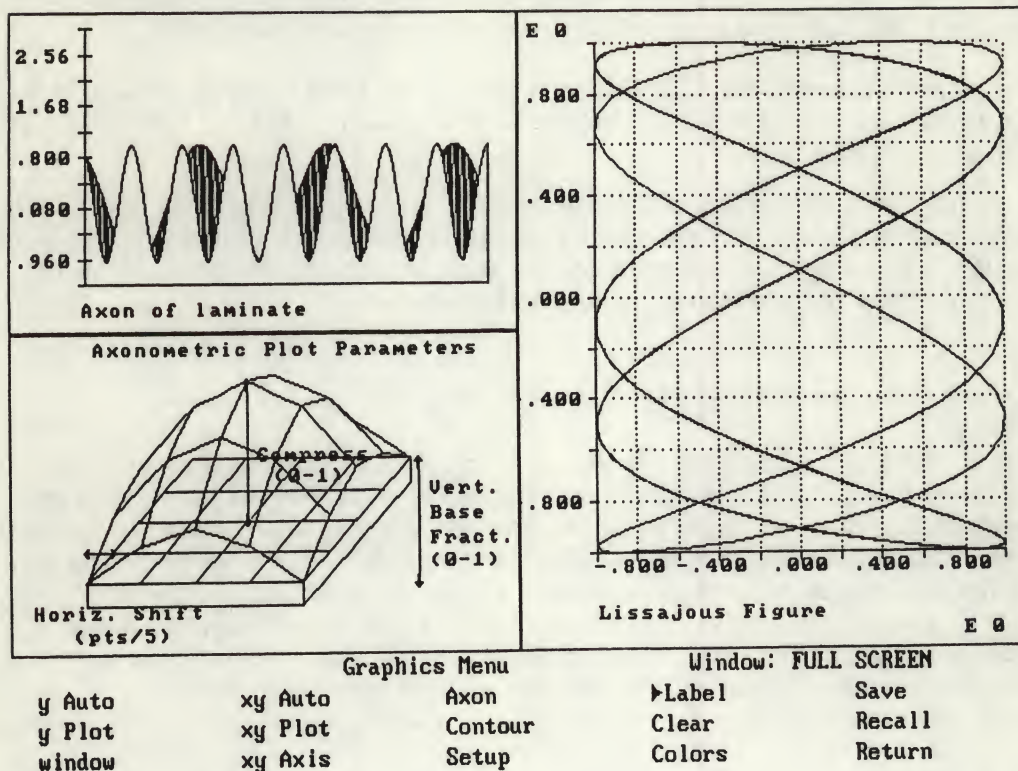
PgDn

Switches the display to the previous function. If the current function has been modified it forces a recompilation of the command lines.

Esc

Exits from the **Functions** menu option and returns to the main menu. If the current function has been modified it is recompiled before exiting **Functions**

CHAPTER 4: GRAPHICS



4.1 Overview

4.1.1 GENERAL DESCRIPTION

The Graphics menu is used to generate graphical representations of the data arrays stored in the system Variables, R-Z. Graphs, or plots, may be drawn on the display or on a plotter (see Chapter 1, Section 1.2.3 and 1.2.2 for display and plotter configuration specifications). Graphs may also be saved in disk files. The user may draw in any of nine windows that subdivide the plot area: four quadrants, four halves (top, bottom, right and left), or the full screen.

The Graphics option is very useful as a preliminary scratchpad when drawing graphs on a plotter. All plot parameter options, such as Window, Label, Colors, and Setup, can be quickly and interactively modified on the display before producing the final penned plot.

4.1.2 CARTESIAN PLOTS

Cartesian plots are formed using a single array of y-values (**y Auto** or **y Plot**) or using arrays for both the x- and y-values (**xy Auto**, **xy Plot**). If only the y-array is used, the x-values are taken to be the array indices. For cartesian plots, the data must be one dimensional and may be read from either a one dimensional array or from a row of a two dimensional array.

Cartesian plots may be displayed with or without automatic sizing. When the automatic sizing feature is applied (i.e., when the options **y Auto**, **xy Auto** are used) the current window is cleared and a new pair of axes are displayed. The ranges of these axes are automatically scaled to optimize display of the selected data. The data is then plotted on the new coordinate axes.

When automatic sizing is not used (**y Plot**, **xy Plot**) the plot is superimposed on the contents of the current display window, using whatever axes ranges are currently in effect.

The **Setup** option is used to manually set each axis' range and select the scale and plot style. For example, axes can be displayed with or without grids, scales can be linear or logarithmic, and each axis can be repositioned on the screen. Each window can have its own unique **Setup**.

TIP

To display two or more plots in the same graph it is often convenient to plot the first one using the automatic sizing feature and then plot the subsequent ones without it. The first **Auto** plot will clear the window before drawing the scaled axes and graph. The subsequent **Plots** will be superimposed on the first graph. If the axis ranges selected by automatic sizing are unacceptable for the subsequent plots (an "invalid 8087 operation" error may result), use **Setup** to explicitly specify the axis ranges, followed by **xy Axis** to display the new coordinate axes. Then redraw the plots without automatic sizing.

4.1.3 TWO DIMENSIONAL REPRESENTATIONS

The menu selections **Axon** and **Contour** generate, respectively, axonometric and contour plots of two dimensional arrays.

4.1.4 OTHER FEATURES

Two additional Graphics display features provide the means to interactively position labels on a graph (see **Label**) and to select the plot, axes, label, and background colors (see **Colors**). The color assignments are associated with the current graphics window. When a window is selected for the first time, the most recent color assignments are used until they are reset.

4.2 The Graphics Menu

When a plot option is selected from the Graphics Menu, the Variables R-Z are scanned to see if a suitable array is available. If none are found, an error message appears on the screen.

y Auto

Plots an array of y-values in the current graphics window using an automatic sizing routine to choose ranges for the x and y axes. The array indices form the x component and the array values form the y component of each (x,y) pair. The window is cleared and the axes are drawn using the plot style specified in **Setup**. The plot is then drawn on the new axes.

On selecting **y Auto**, the user is prompted for the Variable array to plot. If a two dimensional array is specified a second prompt requests the user to specify which row to plot (only vectors are allowed on a cartesian plot). **Esc** accepts the default row. Once the array and row (if necessary) are specified, **Esc** enters the following submenu of **y Auto** operations:

Display Graph	-- Draw graph in the current window
To Plotter	-- Draw graph on the plotter
New Parm	-- Select new array or row parameters
Return	-- Return to the Graphics menu

y Plot

Plots an array of y-values in the current window using the current axes ranges to scale the plot (rather than automatically scaling the axes to the data range as **y auto** does). The array indices form the x component and the array values form the y component of each (x,y) pair. The plot is superimposed on the contents of the current window.

On selecting **y Plot** the user is prompted for the Variable array to plot as in **y Auto**. Once the array and row (if necessary) are specified, **Esc** enters the following submenu of **y Plot** operations:

Display Graph	-- Draw graph in the current window
Erase Graph	-- Erase last graph drawn
To Plotter	-- Draw graph on the plotter
New Parm	-- Select new array or row parameters
Return	-- Return to the main menu

Window

Selects one of nine graphic windows as the current output window. **Setup** parameters and **Color** assignments are associated with each graphics window, so when the current window is changed, the setup and colors change with it. When a window is selected for the first time, the default setup parameters and color assignments are used.

A submenu is displayed containing nine window region choices. After selecting a window, the user is automatically returned to the Graphics menu. Subsequent plots will be directed to the new window.

xyAuto

Plots a pair of x and y arrays in the current graphics window using an automatic sizing routine to choose ranges for the x and y axes. The window is cleared (when drawing on the display) and the axes are drawn using the plot style specified in **Setup**. The plot is then drawn on the new axes.

On selecting **xy Auto**, the user is prompted for the Variable arrays to plot as the x- and y-arrays. (The same Variable array may be used for both). If a two dimensional array is specified a second prompt requests the user to specify which row to plot. Enter the row number or simply press **Enter** to accept the default. If you wish to subsequently change the row number, you must respecify the Variable array. Once the arrays and rows (if necessary) are specified, **Esc** enters the following submenu of **xy Auto** operations:

Display Graph	-- Draw graph in the current window
To Plotter	-- Draw graph on the plotter
New Parm	-- Select new array or row parameters
Return	-- Return to the Graphics menu

xy Plot

Plots a pair of x and y arrays in the current graphics window using the current axes ranges to scale the plot. The graph is superimposed on the current contents of the graphics window.

On selecting **xy Plot**, the user is prompted for the Variable arrays to plot as the x- and y-arrays, as in **xy Auto**. Once the arrays and rows (if necessary) are specified, **Esc** enters the following submenu of **xy Plot** operations:

Display Graph	-- Draw graph in the current window
Erase Graph	-- Erase last graph drawn
To Plotter	-- Draw graph on the plotter
New Parm	-- Select new array or row parameters
Return	-- Return to the main menu

xy Axis

Clears and plots the xy axes in the current window. The axis range and style are specified by the current **Setup** parameters. The xy Axis submenu contains the following options:

Display Axes	-- Draw axes in the current window
To Plotter	-- Draw axes on the plotter
Return	-- Return to the main menu

Axon

Draws an axonometric plot of a two dimensional array. The plot is displayed with hidden lines removed.

On selecting **Axon**, the user is prompted for the two dimensional Variable array to plot and for specification of the following three plot parameters:

Amplitude Compression Factor

Adjusts the scale of the vertical axis. Increasing the compression factor will cause the range of the vertical axis to be compressed, thus increasing the span of data variations. The compression factor can range between 0 and 1.

Vertical Base Fraction

This fraction refers to the portion of the window that falls between the baselines for the plots of the first and last rows. Increasing this factor causes the plot to appear as if the data is being viewed from a higher elevation. The values for this parameter must also range between 0 and 1.

Horizontal Shift

This parameter controls the skew of the plot by specifying the horizontal shift (in units of $1/5$ of a data point) between the baselines of the plots of successive rows. For example, if the horizontal shift is 5, then the first data point in the second row will be plotted on the same vertical line as the second data point in the first row. The value for this parameter must be an integer ranging from 0 to 20.

Once the prompt responses are entered (or left in their default state) the **Esc** key records the values and displays the following submenu of **Axon** operations:

Display Graph	-- Draw the axonometric plot in the current window
To Plotter	-- Draw the axonometric plot on the plotter
New Parm	-- Select new array or plot parameters
Parm Display	-- Draw a plot in the current window illustrating the significance of the three plot parameters
Return	-- Return to the Graphics menu

Contour

Draws a contour plot of a two dimensional array. On entry, the user is prompted for the two dimensional Variable array and the number of contours (up to 12) to plot. The contour levels are chosen to be equally spaced between the minimum and maximum data values. For example, if two contours are requested on an array of data values ranging from 0 to 1, the contour levels will be set at 1/3 and 2/3.

If an Enhanced Color/Graphics Adapter (EGA) card is used, the contours will be plotted in different colors. When 16 colors are available (i.e., Mode 1: EGA card with more than 64K bytes of memory driving an Enhanced Color Display) the contours are assigned successive colors starting with 1 (refer to the color charts under **Colors** below). In that case the window (viewport) color should be chosen to be either larger than the number of contours or 0. If only 4 colors are available (Mode 2), then the contour colors will be assigned according to the repeating sequence 1,2,3,1,2,3,..., and the window color should be set equal to 0. Use **Colors** to set the window color.

The **Esc** key accepts the entered (or default) values and displays the following submenu of **Contour** operations:

Display Graph	-- Draw the contour plot on the display
Display Key	-- Display a key to the contour levels
To Plotter	-- Draw the contour plot on the plotter
New Parm	-- Select new array or plot parameters
Return	-- Return to the Graphics menu

Setup

Sets axes parameters. On entry, a prompt list is displayed for the x-axis with its current parameter values as defaults. The first prompt is used to toggle between the parameter lists for each axis.

Setup parameters are associated with each graphics window; when the current window is changed, the setup parameters change with it. When a window is selected for the first time, the default setup parameters are used. The parameters are listed below.

X or Y Parameter	-- Toggle the axis selection
Min	-- Set minimum value for axis range (only positive values are accepted for logarithmic scaling)
Max	-- Set maximum value for axis range (only positive values are accepted for logarithmic scaling)
Log (Y or N)	-- Toggle between linear and logarithmic axis scaling. For Log scaling the axis range is reset to the interval [1,100].
Labels (Y or N) display on and off	-- Toggle axis numerical label
Axis (Y or N)	-- Toggle axis display on and off
Grid (Y or N)	-- Toggle grid display on and off
Origin (B/M/T) (R/M/L)	-- Position axis in the window x-axis: bottom, middle, top y-axis: right, middle, left

Label

Inserts textual labels in the current graphics window. On entry, the user is prompted for the label name, label direction, and character direction. The two directions are specified in degrees. When drawing on the display, the label direction must be a multiple of 45 degrees and the character direction a multiple of 90. If other values are chosen, the direction will default to 0 degrees when drawn. When drawing on a plotter, the directions may assume any values.

Esc accepts the prompt responses (or the default when no response is given) and displays the following submenu of **Label** operations:

- Locate** -- Select the label position. A cursor appears on the current plot and is controlled with the following keys:
 - Arrows** - reposition the cursor
 - PgUp n** - set the cursor movement step size to the number **n** (typed immediately after **PgUp** is pressed). **n** may be any value. The full display is scaled to 100x100 so with a step value of 25 the cursor would move a quarter screen length when an **arrow** key is pressed.
 - Del** - mark cursor position and return to the **Label** submenu.
- Display** -- Display the label in the current window starting at the marked position
- Erase** -- Erase the last label displayed (labels displayed in previous **Label** sessions cannot be erased)
- To Plotter** -- Print the label on the plotter starting at the marked position
- New Label** -- Select new label or direction parameters
- Return** -- Return to the Graphics menu

Note: Textual labels take on the "plot" color, defined in the **Colors** option, not the "label" color. The "label" color refers to the numerical values assigned to the axes.

TIP

Labels are contained within windows. To label a collection of windows on a single screen or plot, expand the **Window** selection to Full Screen before entering the **Label** option. Labels may then be placed anywhere on the display.

Clear

Clears the current display window.

Colors

Assigns colors to four different plot aspects of the current window. *Colors are only displayed if an Enhanced Color Adapter (EGA) card is in use.* The **Colors** option may be used on any machine, however, to set pen colors for plotters. A small graphics window at the bottom of the screen contains a stylized plot that illustrates the current color assignments (EGA only). This window is updated continuously as the color assignments are modified. Color assignments are associated with each window; when the current window is changed, the colors change with it. When a window is selected for the first time, the default color assignments are used. The plot aspects listed in the **Colors** submenu are:

Axes	-- Set color of axes
Label	-- Set color of numerical axes values
Vuport	-- Set background color of window (display only)
Plot	-- Set color of plot and textual labels

Color values correspond to display colors as follows ("Mode" is the display hardware selection number on the System Configuration menu -- see Chapter 1):

Value	Mode 1	Mode 2	Mode 3	Mode 4
0	Black	Black	Black	Black
1	Blue	Cyan	Blue	Cyan
2	Green	Red	Green	Red
3	Cyan	White	Cyan	White
4	Red	Cyan	Blue	Cyan
5	Magenta	Red	Green	Red
6	Brown	White	Cyan	White
7	Lt.Grey	Cyan	Blue	Cyan
8	Dk.Grey	Red	Green	Red
9	Lt.Blue	White	Cyan	White
10	Lt.Grn	Cyan	Blue	Cyan
11	Lt.Cyan	Red	Green	Red
12	Lt.Red	White	Cyan	White
13	Lt.Mag	Cyan	Blue	Cyan
14	Yellow	Red	Green	Red
15	White	White	Cyan	White

When drawing on a plotter, pen numbers are equated with the color values. For example, if the axes color is set to 1, a prompt for pen #1 will appear before the axes are plotted.

Note: 0 is an invalid pen number. When 0 is used as a color value, a prompt for pen #16 will be displayed instead.

Save

Copies the contents of the current graphics window into a disk file. The size of the file depends on the size of the current window as well as the graphics mode of the display. A full sized window on an EGA display with more than 64K bytes of memory is saved in a 96K byte file. (See Table R1.1, section 1.2.1).

TIP

Save copies only the current window to disk. To save several windows, expand the Window to Full Screen before executing Save.

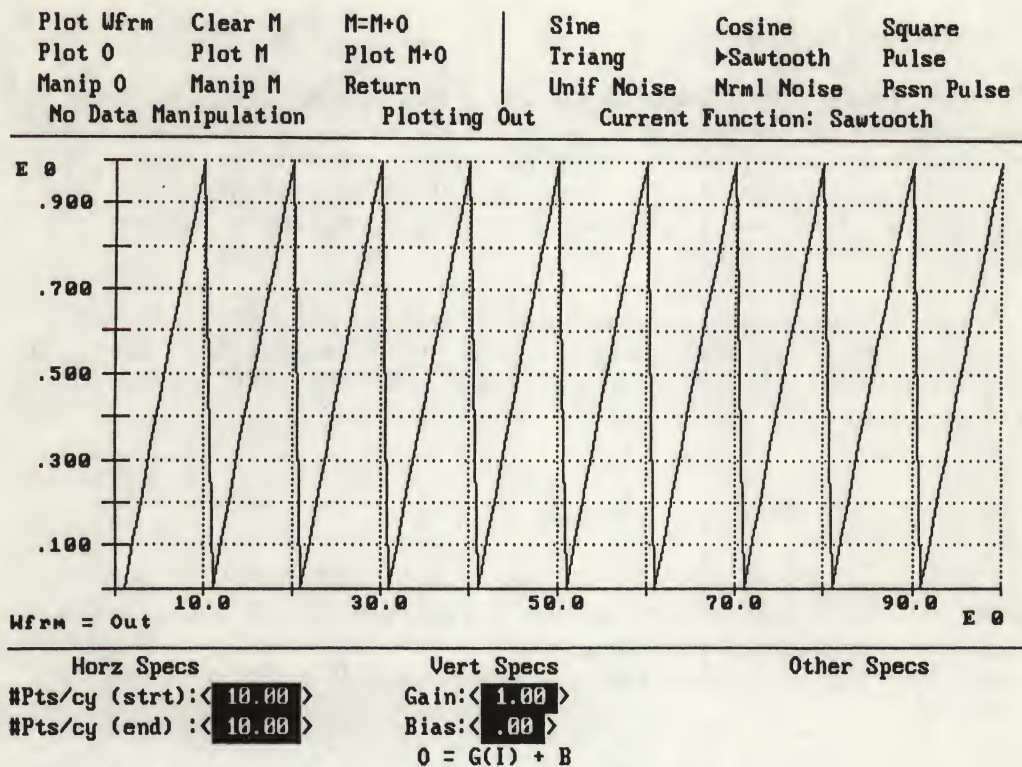
Recall

Recalls a graphics window from disk into the current window. *The current window must be the same shape as the one that is being recalled.*

Return

Returns processing to the Desk Calculator menu.

CHAPTER 5: WAVEFORM GENERATOR



SCREEN R5.1

5.1 Overview

5.1.1 GENERAL DESCRIPTION

The Waveform Generator (Wave Gen) provides a convenient, interactive means to create periodic and noise waveforms for use in simulations, testing of processing procedures, and with the ASYSTANT+ as a D/A source.

Wave Gen emulates an analog function generator, providing many similar controls -- waveform type, gain, bias, and frequency. However, Wave Gen does not produce time-varying voltages, but rather arrays of digital values which can

be saved in Desk Calculator Variables for further processing or output. The waveform types include a standard set of periodic functions (eg., sinusoids, square, and triangular) and various types of noise (eg., white Gaussian and Poisson impulses). A linear frequency sweep can be applied to periodic waveforms.

In addition to outputting waveforms, the generator can also store waveforms in memory. This allows you to create composite waveforms by cumulatively summing multiple outputs into memory eg., to create a sinusoidal waveform with noise. For complete flexibility, this memory can be manipulated through the Mini Calculator.

5.1.2 ENTERING WAVE GEN

Immediately after selecting Wave Gen from the Main Menu, you will be prompted to enter the number of points for the waveforms generated and for an initial random number seed. The default seed is selected from the low order digits of your PC's clock and is therefore not likely to be repeated by coincidence. You need to set it manually **only** if you wish to replicate a pseudo-random waveform.

The **Quit/Generate** prompt is the entry and exit point of Wave Gen: **Generate** enters Wave Gen and **Quit** returns to the Main Menu. Wave Gen's controls will be set to defaults on first entry, and subsequently to the state you left them in.

5.1.3 OUTPUT AND MEMORY WAVEFORMS

Wave Gen uses two waveforms: the output waveform and the memory waveform. Storage in memory works like a Memory+ button on a pocket calculator: the output waveform is added to the current contents of the memory waveform. The memory waveform is initialized to 0, and may be cleared manually.

From within Wave Gen, the memory waveform can be pushed onto the stack by executing the command **MEM** in the Mini Calculator. Any calculator operation can then be performed. The complementary word **>MEM** is available to remove an array from the Mini Calculator stack and place it in the memory waveform. The only restrictions on the use of **>MEM** are that the array:

1. must have an identical number of points as the output waveform
2. must be one-dimensional, and
3. must not be complex.

The combination of **MEM**, **>MEM**, and the Wave Gen functions provides the means to create essentially any type of waveform.

Throughout the Wave Gen menus, the output waveform is designated by the letter **O** and the memory waveform by the letter **M**.

NOTE

The command words **MEM** and **>MEM** are only available from within Wave Proc (Chapter 6) and Wave Gen. (The Wave Proc and Wave Gen memories are independent and do not interact with each other in any way). Memory contents will be cleared when either option is exited and the words will have no meaning outside of their context.

5.1.4 THE MENUS

Four menus are available in the Waveform Generator. Three are visible in normal operation: the Main menu, the Type menu, and the Horizontal/Vertical Specifications menu. The fourth is the Manipulation menu and is opened by selecting **Manip O** or **Manip M** in the Main menu.

The **Main menu** appears in the upper left corner of the screen and allows you to toggle plotting options, plot the waveform, add the output waveform to the waveform in memory, clear the waveform in memory, or enter the Manipulation menu.

The **Type menu** is located in the upper right corner of the screen and sets the type of waveform to be output. Periodic and noise waveforms are available.

Noise waveforms are created using a random number generator. Like all computer random number generators, this computes a deterministic series of numbers starting from an initial seed (random number). It will appear random - even to rigorous analysis. However, the same series will always be produced if the seed is the same. Consequently, you may duplicate a noise waveform by using the same initial seed again. Alternatively, a different noise waveform will be generated if a new initial seed is used.

Whenever the **ASYSTANT** is entered from DOS, the initial seed is determined from an algorithm applied to the time of day in milliseconds. Thus, unless you enter the **ASYSTANT** at the same millisecond each day of use, you are assured of a new initial seed. However, as previously stated, it is possible to set the initial seed on entry to the Wave Generator menu, thus providing a means of reproducing random waveforms exactly.

The third menu is the **Horizontal/Vertical Specifications** menu and is located on the bottom of the screen. The menu options (**Horz Specs**, **Vert Specs**, and **Other Specs**) emulate the "knobs" on a hardware function generator that control the period, gain, bias, and other features of the output waveform. A prompt list is opened upon selection of these menu items; values entered into the prompts change the horizontal and vertical nature of the output waveform.

The **Manipulation** menu overwrites the Horizontal/Vertical menu and is displayed only when the **Manip O** or **Manip M** option in the Main Menu is selected. It lets you manipulate the output and memory waveforms as arrays of data, providing extra flexibility. You can **Reverse**, **Invert**, and **Rotate**, or use a graphical **Editor** to create totally arbitrary waveforms.

5.1.5 SPECIAL KEYS AND STATUS

Movement between menus is controlled by the **PgUp**, **PgDn**, and **Tab** keys: **Tab** toggles between the Main menu on the top left and the Waveform Type menu on the top right. **PgUp** and **PgDn** toggles between the bottom Horizontal/Vertical Specifications menu and the current top menu.

Manipulation status, plotting status, and current function are displayed on the Status line directly above the plotting area.

5.1.6 EXITING THE WAVEFORM GENERATOR

When **Return** is selected on the Wave Gen Main menu a prompt list appears [refer to Screen R5.2] presenting three options:

Generate

Re-enters Wave Gen. You may first change the number of points in the waveform and the random number seed. If you change the number of points, the output and memory waveforms will be cleared, but the controls will remain the same as when you left.

Save

Saves the output and memory waveforms on the stack or in Variables. Choices are made in the third and fourth prompts. Enter **#** to save the waveform on the stack or a letter **R-Z** to save it in a Variable. The default **D** for Delete will not save the waveform. You may return to Wave Gen after executing **Save** by selecting **Generate**.

Quit&save

Saves the output and memory waveforms and exits to the Desk Calculator. The waveforms may be saved on the stack or in Variables; choices are made in the third and fourth prompts as in **Save** above.

Note: The two save prompts default to **D** for Delete which specifies that the waveforms are *not to be saved*. Remember that all data in the Wave Gen repositories are erased and the memory waveform will no longer be accessible with the command word **MEM** once you exit Wave Gen. Therefore, be careful to save any waveforms you think you might need.

Stack Contents	Parameters
Unnamed Array REAL DIM[400]	A= 1.0000 B= 5.0000 C= .2000 D= 1.0000 E= 2.0000 F= .5000 G= .0000 H= .0000 I= .0000
Wave Generator	Variables
Number of waveform points: (100) Random noise seed: (58944) Save in R - Z, # or Delete -- Out: (W) Mem: (X) Save/Quit&save/Generate: * Q *	EKGdata R= 400 REAL BASIC S= 200 REAL 1-2-3 T= 200 REAL U= 111 REAL V= 111 REAL W= .0000 X= .0000 Y= .0000 Z= .0000

SCREEN R5.2

5.2 Main Menu Description

5.2.1 GENERAL DESCRIPTION

The Waveform Generator Main menu, located in the upper left window, is used to control plotting options, to clear or add to the memory waveform, and to enter the Manipulation menu.

Note: Plot O, Plot M, and Plot M+O only specify which waveform is to be acted on. Plot Wfrm actually plots the array.

Each option is described below:

Plot Wfrm

Plots the currently selected waveform: Output (O), Memory (M), or Memory + Output (O+M). It is useful to generate a new display immediately after changing the plotting option.

Clear M

Clears the memory waveform to zero. The current output waveform is unaffected.

M=M+O

Adds the current output waveform to the memory waveform to create a composite in memory. The output waveform will be cleared to zero. On entry into Wave Gen, memory is initialized to zero.

Plot O

Toggles the plotting option to plot the current output waveform. A new plot is generated after every operation affecting O.

Plot M

Toggles the plotting option to plot the current memory waveform. A new plot is generated after every operation affecting M.

Plot M+O

Toggles the plotting option to plot the sum of the current output waveform and the current memory waveform. This lets you view the sum of the output and memory waveforms without affecting the contents of the memory waveform.

TIP:

Remember, **Plot M+O** does not alter the contents of memory. If you wish to save M+O, remember to execute **M=M+O** before exiting Wave Gen. To be safe, save both M and O in calculator Variables on exit. If you subsequently discover you forgot the final summing operation, you can easily add the two Variables in the Desk Calculator.

Manip O

Opens the manipulation menu for the output waveform. A detailed description of this menu is given in Section 5.5.

Manip M

Opens the manipulation menu for the memory waveform. A detailed description of this menu is given in Section 5.5.

Return

Exits Wave Gen. A prompt list opens allowing the output and memory waveforms to be saved in Desk Calculator Variables or on the stack. You may then reenter the Waveform Generator or quit.

Note: On occasion, Wave Gen will automatically toggle the plot option so that a meaningful display is always presented. For example, if **Plot M** is in force and a new output waveform is created, the plot option will change to **Plot O**. Similarly, **Plot O** will be changed to **Plot M** when output is added to the memory with **M=M+O**.

5.3 Waveform Type Selection

The Type menu is located in the top right window of the screen. It emulates the switches on hardware function generators to select the output waveform type.

All waveforms are created internally with normalized amplitudes, as described below. Frequency, Gain, and Bias are options on the Horizontal/Vertical menu at the bottom of the screen. Some types of waveforms also let you use an **Other Specs** option for adjustment. The Horizontal/Vertical menu is described in Section 5.4.

Sine

Creates a sine wave with a normalized amplitude between -1 and +1.

Cosine

Creates a cosine wave with a normalized amplitude between -1 and +1.

Square

Creates a square wave with a normalized amplitude between -1 and +1.

Triang

Creates a triangle wave with a normalized amplitude between 0 and +1.

Sawtooth

Creates a sawtooth wave with a normalized amplitude between 0 and +1.

Pulse

Creates a pulse wave with a normalized amplitude between 0 and +1. The random number generator is used to vary the pulse amplitude with a normal distribution so that each selection will cause a (randomly) different waveform to be generated (unless the random noise seed is explicitly set). A prompt list of additional specifications is created under the **Other Specs** menu item (see Section 5.4).

Unif Noise

Creates white noise with a uniform distribution between -1 and +1. This uses the random number generator. Each selection will cause a (randomly) different waveform to be generated (unless the random noise seed is explicitly set). A prompt list of additional specifications is created under the **Other Specs** menu item (see Section 5.4).

Nrml Noise

Creates white noise with a normal (Gaussian) distribution of mean 0 and variance 1. This uses the random number generator. Each selection will cause a (randomly) different waveform to be generated (unless the random noise seed is explicitly set.) A prompt list of additional specifications is created under the **Other Specs** menu item (see Section 5.4).

Pssn Pulse

Creates a Poisson impulse train in which the delay between adjacent impulses follows an exponential distribution. The average delay defaults to approximately 1/10 the number of points in the waveform. The impulse amplitude varies randomly with a normal (Gaussian) probability distribution with a default distribution of unity mean and zero variance (all pulses have equal height of one). This uses the random number generator. Each selection will cause a (randomly) different waveform to be generated (unless the random noise seed is explicitly set). A prompt list of additional specifications is created under the **Other Specs** menu item (see Section 5.4).

Note: The randomization of waveforms, such as the noise waveforms, will remain constant when manipulated and/or replotted, since they are recalculated using the same random seed. The seed is updated only when a new selection is made from the Waveform Type menu or when Wave Gen is reentered.

5.4 Horizontal and Vertical Specifications

The Horizontal/Vertical Specifications menu is used to tailor the selected waveform type for frequency, amplitude, bias, and other parameters. The menu occupies the bottom screen window and is entered and exited by pressing the **PgUp** and **PgDn** keys. Three options control the horizontal attributes (**Horz Specs**), the vertical attributes (**Vert Specs**), and other special attributes (**Other Specs**). Each option opens a prompt list for adjusting the output waveform settings. **Esc** exits the prompt list, calculates a new waveform using the new settings, and automatically replots the output waveform.

Noise waveforms are not periodic. Consequently, the **Horz Specs** menu item is not available for noise waveforms.

Horz Specs

Opens a prompt list for horizontal specifications. Waveform cycles are specified in points per cycle, rather than frequency. Linear frequency sweeps are created by specifying a different number of points per cycle at the beginning of the waveform than that at the end. The start (end) frequency is the reciprocal of the start (end) period. The sweep can be set for either decreasing or increasing frequency. These specifications are not available for noise waveforms.

Vert Specs

Opens a prompt list for specification of waveform gain and bias. Note that gain and bias adjustments do not interact. For example, a change in gain does not reposition the bias point of the waveform. Computation follows the formula: $O = G(I) + B$, or output equals gain times normalized input plus bias.

Other Specs

Opens a prompt list of special parameters which can be used to tailor the horizontal and vertical nature of randomized waveforms. Different parameters are provided for each type of waveform. The parameters associated with waveform types that use **Other Specs** are given below:

Pulse

Average amplitude, amplitude standard deviation, and % duty cycle.

Unif Noise

Average amplitude and amplitude span.

Nrml Noise

Average amplitude and amplitude standard deviation.

Pssn Pulse

Average delay between pulses (in points), average amplitude, and amplitude standard deviation.

If the waveform type does not support special attributes, no prompts will be shown below the **Other Specs** item.

5.5 The Waveform Manipulation Menu

The output and memory waveforms can be processed as data arrays using the Waveform Manipulation Menu. The menu is entered from the Main Wave Gen menu by selecting either **Manip O**, to manipulate the output waveform, or **Manip M**, for the memory waveform. This supplements the capabilities of the Horizontal/Vertical menu and even allows creation of arbitrary waveforms by using a graphical editor. The manipulation menu is usually more convenient than the Mini Calculator because it is menu driven and provides immediate graphical feedback. It is, however, significantly more limited.

Manipulation will be done only on the specified waveform, regardless of plot option. However, the Wave Generator will automatically change the current plot option to ensure that the manipulated waveform will be plotted in some form. For example, if the memory waveform is manipulated and **Plot O** is currently selected, **Plot M** will automatically be selected.

Min/Max

Opens a prompt list to allow specification of the minimum and maximum. When **Esc** is pressed, the waveform will be scaled to span the range between the specified minimum and maximum.

Reverse

The beginning of the waveform becomes the end.

Invert

Same as multiplying by -1.

Rotate

Opens a prompt list used to specify a desired waveform shift (in points). The shift is "circular" so that the end of the waveform wraps around to the beginning. Positive values rotate the waveform to the right. The shift occurs when you exit the prompt by pressing **Esc**. This option is useful for changing the starting phase of periodic waveforms.

Editor

Sets up the cursor keys to graphically edit the waveform. Two marked points can be located arbitrarily on the screen. The original data lying in the horizontal section between these two points are replaced by a straight line connecting the two marked points. Multiple linear sections can be catenated to create an arbitrary waveform.

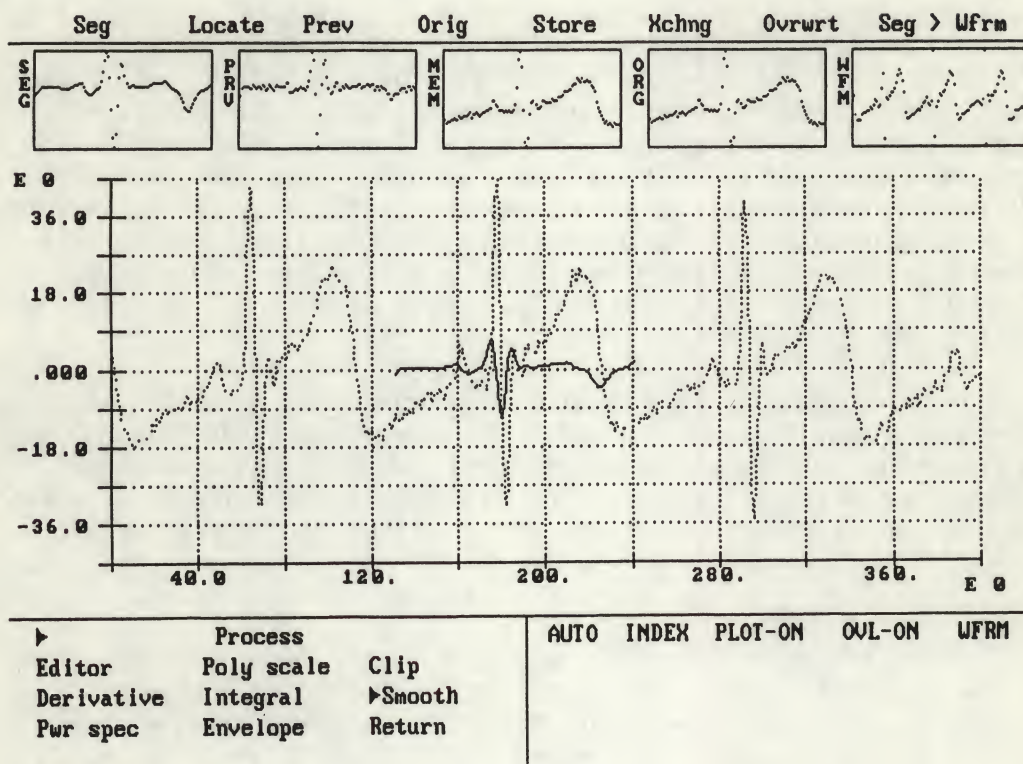
During editing, only the currently manipulated waveform is displayed. If **Plot M+O** was in effect on entry, it will be suspended until the editor is exited.

The motion of the graphics cursor is controlled by the **arrow** keys. The cursor moves along the waveform with the **left/right arrow** keys. The **up/down arrow** keys move the cursor off the waveform. If the cursor is off the waveform, any use of the **left/right arrow** keys will return the cursor to the waveform. This procedure assures that the cursor is always located directly above, on, or beneath an actual data point.

The numeric pad keys have the following functions:

- | | |
|--------------------|---|
| Up arrow | -- Moves the graphics cursor above the waveform. |
| Down arrow | -- Moves the graphics cursor below the waveform. |
| Right arrow | -- Moves the graphics cursor right along the waveform. |
| Left arrow | -- Moves the graphics cursor left along the waveform. |
| Ins | -- Marks a point. If this is the first point, an asterisk will be displayed at the cursor position. If it is the second point, a linear interpolation will be done between the two marked points. |
| Del | -- Erases the first marked point, in case of a mistake. |
| PgUp | -- Increases the distance the graphics cursor moves along the waveform with every use of the left/right arrow keys. |
| PgDn | -- Decreases the distance the graphics cursor moves along the waveform with every use of the left/right arrow keys. Minimum motion is one data point per keystroke. |
| Esc | -- Returns to the Waveform Manipulation Menu. |

CHAPTER 6: **WAVEFORM PROCESSOR**



SCREEN R6.1

6.1 Overview

6.1.1 GENERAL DESCRIPTION

The Wave Processor (Wave Proc) is used for waveform processing and analysis in an interactive, graphics-based environment. It operates as a "waveform calculator" on single, real, one-dimensional arrays (waveforms) and provides automatic display and storage of intermediate results. Although its processing capabilities are more limited than the Desk Calculator and the File Processor, its graphics interface makes it the environment of choice for many applications.

6.1.2 ENTERING THE WAVEFORM PROCESSOR

On entry, Wave Proc scans the top stack entry and the Variables for an array. The entry prompt will default to the first array found. The user can accept the default or designate another array (# is used to designate the top stack entry). If the selected input is a two-dimensional array, Wave Proc will request a row number, since only one-dimensional arrays can be processed. You also have the option to **Quit** and return to the Desk Calculator, or continue by choosing the **Process** option.

6.1.3 WAVEFORM AND SEGMENTS

It is often desirable to process one portion of a waveform differently from another. For example, a waveform might need extensive smoothing in one portion and only light smoothing elsewhere. Wave Proc lets you visually locate a region of the input waveform for processing or analysis by using **Locate** (for **Locate Segment**) in the **Seg** menu at the top of the screen. The location of a segment is specified by its left and right endpoints. This can be specified numerically, by entering numbers corresponding to the endpoints, or graphically, by scrolling through the waveform and using cursor marks to designate the segment boundaries. Marked data is placed in a repository and termed the **Current Segment (SEG for short)**; **SEG** is used in all subsequent processing and analysis steps.

The Waveform Processor maintains several other data repositories which allow you to store and recall "images" of the current segment in different stages of processing. All repositories are initialized with the complete waveform specified on entry. The contents of all repositories are displayed at the top of the screen in tiny plots (refer to Screen R6.1). From right to left they are:

WFM

The **Waveform** repository holds the entire waveform originally specified at entry. **WFM** is preserved by the system unless the user explicitly overwrites it. The **Locate** command draws on this repository when the user wishes to redefine the **Current Segment** for selective processing.

ORG

The **Original Segment** repository preserves the segment last selected by **Locate**. In case of a processing accident, the user can restore the **Current Segment** without going back into **Locate**.

MEM

The **Memory Segment** can be used as a repository for intermediate results, much like the memory on a pocket calculator. The **Current Segment** can be stored in, or exchanged with the **Memory Segment**, at any time. The **Memory Segment** can also be accessed through the **Mini Calculator**, as explained below and in Section 6.8.

PRV

Immediately before any processing steps, the Current Segment is copied into the **Previous Segment** repository. This lets you easily restore the Current Segment if an incorrect processing step is performed.

SEG

The **Current Segment** repository stores the waveform used in all processing and analysis steps. The results of processing are returned to this repository.

Note: These plots are displayed to provide quick reference to repository contents; they should not be used to extract quantitative information.

6.1.4 AXIS AND PLOTTING PARAMETERS

Many plotting options are available in Wave Proc. They may be specified from two options, **Set Axes** and **Plot Parms**, found in the Wave Proc Main menu displayed in the lower left window of the screen.

Set axes is used to specify vertical and horizontal axis units (normalized or "real-world") and ranges.

Plot parms provides a set of plotting toggles. For example, you can select to overlay a plot of the memory segment on the plot of the current segment each time a plot is drawn. Or you can toggle plotting on or off altogether. When plotting is off the speed of processing is increased (toggle plotting back on to display the final result).

The current status of the plotting options is indicated on the **Plotting Status line**, located right below the plotting area.

6.1.5 PROCESSING AND ANALYZING THE CURRENT SEGMENT

Separate menus enable processing and analysis to be performed on SEG, the waveform contained in the current segment repository. These menus are invoked by the **Process** and **Analysis** options of the Wave Proc Main menu.

Process menu contains various methods to process SEG, such as **Smooth** and **Integrate**. Each operation in **Process** will alter the contents of SEG. As a consequence, the PRV repository is updated each time a **Process** operation is selected.

The **Analysis** menu contains some common methods of extracting quantitative data from your waveform, for example, **Rise time** and **Peak width**. The numbers generated in this menu can be printed or saved on disk using

Note #'s. Analysis operations are descriptive only, and will not change the contents of SEG. PRV is therefore not updated either.

6.1.6 MEMORY OPERATIONS

Arithmetic operations can be performed between SEG and MEM using the **Memory Ops** option. For instance, MEM can be subtracted from SEG by selecting **Seg - Mem**. **Memory Ops** also contains options for storing SEG in MEM or exchanging their contents.

6.1.7 THE MEMORY SEGMENT AND THE MINI CALCULATOR

In addition to the operations provided in the **Memory Ops** menu, MEM can be accessed through the Mini Calculator. This provides a link to the processing features of the Desk Calculator and greatly expands the capabilities of Wave Proc. Hitting the space bar will invoke the Mini Calculator at any time. You can then push the contents of the memory segment on the stack by entering the command word MEM. Once on the stack, you can process the waveform segment with Desk Calculator commands, and/or store it in a Variable.

You may also replace the contents of MEM with an array on the stack by using the command **>MEM**. **>MEM** requires that the data array be non-complex and of the same size and shape as the memory segment, i.e., a one-dimensional real array with the appropriate number of data points.

The ability to exchange SEG and MEM within Wave Proc, and the stack and Variables within the Mini Calculator, offers virtually unlimited processing possibilities.

NOTE

The command words MEM and >MEM are only available from within Wave Proc and Wave Gen (Chapter 5). (The Wave Proc and Wave Gen memories are independent and do not interact with each other in any way). Memory contents will be cleared when either option is exited and the words will have no meaning outside of their context.

6.1.8 EXITING THE WAVEFORM PROCESSOR

Wave Proc is exited by selecting **Return** in the Main menu. A prompt list (see Screen R6.2) will open, along with the current display of Desk Calculator Variables. The prompts gives you three choices:

Process

Re-enters Wave Proc exactly as you left it.

Save

Saves the segment repositories in Variables. Simply enter the appropriate Variable letter next to the segment prompt. Arrays will be saved when the Save option is executed. You can then return to the Wave Proc by executing Process.

Quit&save

Saves the segment repositories in Variables and then exits to the Desk Calculator. Simply enter the appropriate Variable letter next to the segment prompt. Arrays will be saved when the Quit&save option is executed.

Please note: The save prompts default to the letter **D** for Delete, which specifies that the repository is not to be saved when **Quit** is completed. Remember that all Wave Proc repositories will be erased and that the memory segment will no longer be accessible with the command word **MEM** once you exit Wave Proc. Therefore, be careful to save any repositories you think you might need.

Stack Contents	Parameters
Unnamed Array REAL DIM[400]	A= 1.0000
Unnamed Array REAL DIM[111]	B= 5.0000
	C= .2000
	D= 1.0000
	E= 2.0000
	F= .5000
	G= .0000
	H= .0000
	I= .0000
Wave Proc Output Selection	Variables
Save in R,S,T,U,V,W,X,Y,Z,# (stack) or Delete	EKGdata R= 400 REAL
Current Segment : (U)	BASIC S= 200 REAL
Memory Segment : (V)	1-2-3 T= 200 REAL
Current Waveform: (#)	U= .0000
	V= .0000
	W= .0000
	X= .0000
	Y= .0000
	Z= .0000
Save/Quit&save/Process: * S *	

SCREEN R6.2

6.2 The Waveform Processor Main Menu

The Wave Proc Main menu is simply a hub from which you can access all features of the Waveform Processor.

Process

Opens the **Process** menu which contains routines to be used for processing the current segment. See Section 6.6 for details.

Analysis

Opens the **Analysis** menu which allows you to analyze the current segment. See Section 6.7 for details.

Memory Ops

Opens a menu of arithmetic operations to perform on the current and memory segments. See Section 6.8 for details.

Plot parms

Opens a menu of plotting parameters. See Section 6.4 for details.

Set axes

Provides a menu to set the vertical and horizontal axis parameters. See Section 6.5 for details.

Return

Opens a menu to save intermediate results and/or return to the Desk Calculator.

6.3 Manipulating the Segment and Waveform Repositories

The single line of text at the top of the screen is the **Seg menu**. **PgUp** and **PgDn** are used to transfer control between the Segment menu and the current menu at the bottom left of the screen.

The **Seg menu** is used to select a segment from the waveform, overwrite the original data in the waveform with the current segment, and recover from processing mistakes. Each of the **Seg** options is described below:

Locate

Locates a segment within the WFM repository. The location can be done numerically, by entering left and right endpoints in a prompt list, or graphically, by scrolling and marking the endpoints with cursors.

Endpoints can be specified as index or real-world values. The **Index/World** option is referenced for the current setting (see Section 6.5). After the segment is located, the **PRV**, **MEM**, and **ORG** segments are set equal to the current segment, **SEG**.

If you choose to mark the segment graphically, a "scroll and mark" procedure begins. The scroller allows you to move a display window across the waveform. A schematic status line below the plot notes the values of horizontal indices of the marking cursor as they are scrolled across the data.

The numeric pad keys are used to "scroll and mark" the graphic display for various functions in the Waveform Processor, as well as in other Desk Calculator Main Menu options: File I/O and File Processor. The lists below define the use of each key as it is used to scroll and then mark the plot, respectively. The **space bar** is used to toggle between scrolling and marking modes.

SCROLLING KEYS-----

- | | |
|--------------------|---|
| Space Bar | -- Toggle control between scrolling and marking. |
| Left arrow | -- Scroll the display window backward (to the left) through the data. |
| Right arrow | -- Scroll the display window forward (to the right) through the data. |
| Home | -- Position the display window at the beginning of the data. |
| End | -- Position the display window at the end of the data. |

- Ctrl PgUp** -- Position the display window forward by 90% of the window width, leaving the rightmost part of the data in the previous window still displayed.
- Ctrl PgDn** -- Position the display window backward by 90% of the window width, leaving the leftmost part of the data in the previous window still displayed.
- Up arrow** -- Increase the data point increment by a factor of two. A data increment of 4 means that only every fourth point in the data is being displayed. (Beware: since not all data points are being displayed some features of the data may be missed.) A bell is sounded if the data increment cannot be increased.
- Down arrow** -- Decrease the data point increment by a factor of two. The minimum data increment of 1 will display every data point.
- PgUp** -- Increase the horizontal expansion factor by a factor of two, with 8 being the maximum. An expansion of 2 will cause the distance between each displayed data point to be twice that of the normal expansion of one. A bell is sounded if the expansion factor cannot be increased.
- PgDn** -- Decrease the horizontal expansion factor by a factor of two, with 1 being the minimum.
- Del** -- Pause scrolling. Scrolling may be resumed by pressing another scroller key.

MARKING KEYS-----

- Space Bar** -- Toggle control between scrolling and marking.
- Left arrow** -- Move the active mark to the left. The mark may disappear from view, since it can be moved off the screen, but numerical information about the array at the mark location can still be obtained with the **Home** key.
- Right arrow** -- Move the active mark to the right. The mark may disappear from view, since it can be moved off the screen, but numerical information about the array at the mark location can still be obtained with the "Home" key.

- End** -- Make the left mark active.
- PgDn** -- Make the right mark active.
- Up arrow** -- Increase the mark increment by a factor of 2. The number of data points that the mark jumps at each strike of a left or right arrow key equals the mark increment times the data increment.
- Down arrow** -- Decrease the mark increment by a factor of 2, with a minimum of unity. The number of data points that the mark jumps at each strike of a left or right arrow key equals the mark increment times the data increment.
- Home** -- Display the horizontal coordinate (index) of the active mark and the associated value of the array.

Prev

Restores the contents of the PRV repository in SEG, overwriting the SEG contents.

Orig

Replaces SEG with the segment last selected by **Locate**.

Store

Stores SEG in the MEM repository, overwriting the memory contents.

Xchnng

Exchanges the contents of SEG and MEM.

Ovrwrt

Essentially the opposite of **Locate**. Whereas **Locate** copies a segment of data from WFM into SEG, **Ovrwrt** copies the segment in SEG back into WFM at the location specified by the segment endpoints. This option is useful for processing a waveform a segment at a time, and then replacing the original portion with the processed data. **Ovrwrt** does not affect the contents of **ORIG**, **MEM**, **PREV**, or **SEG**.

Seg > Wfrm

Replace WFM with the contents of SEG. This will delete the previous contents of WFM and cause the data in SEG to be treated as the original waveform. **Locate** can now be used on the segment in WFM to specify finer segments.

6.4 Setting the Plotting Options

Plot parms provides a variety of modes for plotting data. The majority allow the SEG to be overlayed with other waveforms in a single plot. Overlays and plotting can be toggled off and on.

The Plotting Status line, located on the lower right below the plotting screen, indicates the current selections of the plotting options.

Prev ovl

Sets the plotting options to overlay SEG with PRV in a single plot, indicated on the Plotting Status line as **PREV**. The horizontal axis will span the range of the current segment (all segments have the same range).

Mem ovl

Sets the plotting options to overlay SEG with MEM in a single plot, indicated on the Plotting Status line as **MEM**. The horizontal axis will span the range of the current segment (all segments have the same range).

Wfrm ovl

Sets the plotting options to overlay SEG with WFM in a single plot, indicated on the Plotting Status line as **WFRM**. Plots will use the horizontal plotting window specified by the **Wfrm window** option in the **Set axes** menu (see Section 6.5).

Ovl tgl

Toggles overlays on and off. The current overlay state is displayed on the Plotting Status line. Toggle overlays off to plot the current segment with maximum use of the plotting area.

Plot tgl

Toggles plotting on and off. The current plotting state is displayed on the Plotting Status line. If **Plot tgl** is on, the data is automatically plotted after each processing step. For routine processing, it may be desirable to turn plotting off to speed up the operation.

Plot data

Plots the data as specified by the current plotting options. **Plot data** ignores the **Plot tgl**.

Scroll

Scrolls the contents of SEG, MEM, or WFM across the display in greater horizontal detail. The horizontal scroll window endpoints and marker location are displayed as indices, if the system is in **Index** mode, or real-world values, if in **World** mode (see Section 6.5). The numeric pad keys function as "scroll and mark" keys, as defined in section 6.3, **Locate**. Quantitative information about the values of the data array can be obtained in marking mode by pressing the **Home** key.

6.5 Selecting Vertical and Horizontal Axis Parameters

The **Set axes** menu is used to configure the vertical and horizontal plotting axes.

The horizontal axis can display the location of data points in two coordinate systems: in array indices, i.e., the first point is labeled 1, the second 2, and so on, or in real-world values, such as seconds. Toggle between these two modes by selecting the **Index/World** option. The Plotting Status line reflects the active mode.

By default, Wave Proc automatically adjusts the plotting range of the vertical axis to accommodate your data. On occasion, it may be more meaningful to specify a fixed vertical range, to keep the zero volt level in view as you monitor changes in the average voltage level (DC) during subsequent processing steps, for example. **Set vert** will open a prompt list to set the vertical axis range. **Auto/Set** lets you toggle freely between the vertical axis range chosen with **Set vert** and an automatically selected range. The Plotting Status line indicates the current setting of **Auto/Set**.

When SEG contains a very small subset of WFM, an overlay of the complete WFM with SEG will produce an unacceptable plot: the features of SEG will be too small to be legible. The **Wfrm window** option allows you to "zoom in" on the segment region for better detail.

An option by option list of features follows:

Wfrm window

Lets you specify the region of WFM to be used in plotting the overlay with SEG. This window is defined by its left and right endpoints. On entry this defaults to the entire input waveform.

Set horiz

Opens a prompt list to specify the minimum and maximum values of the horizontal world and a label for the horizontal axis units.

The default world spans 0 to 1. The default label is "sec". The label can be at most 4 characters long.

Set vert

Allows you to specify the minimum and maximum values of the vertical world. At entry the default world is set to span the minimum and maximum of the input waveform.

?Endpts

Prints the values of the segment and waveform window endpoints in the lower right corner of the screen. These numbers may be saved in the current note file or on the printer by executing Note #'s in the Analysis menu.

Index/World

Toggles between using indices or real-world values for the horizontal axis, indicated on the Plotting Status line as INDEX or WORLD, respectively.

Auto/Set

Toggles between automatic and user-specified selection of the vertical axis range used in data plotting. The state of the vertical axis plotting is indicated on the Plotting Status line as AUTO or SET.

Plot data

Plots the data as specified by the current plotting options (see section 6.4). Plot data ignores Plot tgl.

Scroll

Scrolls the contents of SEG, MEM, or WFM across the display in greater horizontal detail. The horizontal scroll window endpoints and marker location are displayed as indices, if the system is in Index mode, or real-world values, if in World mode (see above). The numeric pad keys function as "scroll and mark" keys, as defined in section 6.3, Locate. Quantitative information about the values of the data array can be obtained in marking mode by pressing the Home key.

6.6 Processing the Current Segment

The **Process** option makes available a set of functions for mathematically processing waveforms. All processing steps operate solely on the contents of **SEG**. After processing, **SEG** will contain the result and the data which was in **SEG** prior to processing will be in **PRV**. The prior contents of **PRV** will be lost. The result is automatically plotted unless **Plot tgl** is off (see section 6.4). The individual processing options are:

Editor

Sets up the cursor keys for graphical editing of a waveform. Data between two arbitrarily placed points on the screen can be replaced by a straight line connecting the two points. The graphics cursor moves along the waveform with the left/right arrow keys. The up/down arrow keys move the cursor off the waveform. Any use of the left/right arrow keys will skip the cursor back to the waveform. This procedure insures that the cursor is always located directly above, on, or beneath an actual data point. Multiple linear sections can be catenated to create an arbitrary waveform.

The numeric pad keys are used as follows:

- | | |
|--------------------|--|
| Up arrow | -- Moves the graphics cursor above the waveform. |
| Down arrow | -- Moves the graphics cursor below the waveform. |
| Right arrow | -- Moves the graphics cursor right along the waveform. |
| Left arrow | -- Moves the graphics cursor left along the waveform. |
| Ins | -- Marks a point. If this is the first point, an asterisk will be displayed at the cursor position. If it is the second point, a linear interpolation will automatically be performed between the two marked points, and the data replotted. |
| Del | -- Erases the first marked point. |
| PgUp | -- Increases the distance the graphics cursor moves along the waveform with every use of the left/right arrow keys. |
| PgDn | -- Decreases the distance the graphics cursor moves along the waveform with every use of the left/right arrow keys. Minimum motion is one data point per key stroke. |

Poly scale

Opens a prompt list for coefficients of up to a 5th degree polynomial. Esc exits the list and causes SEG to be scaled by the specified polynomial. For example, if the power 2 coefficient is set to 1.0, and all other coefficients to 0.0, the output data will be the square of the input.

Clip

Clips SEG to a user-specified minimum and maximum.

Derivative

Performs a derivative on SEG of the order specified (the user is prompted for the order). The derivative is computed by fitting a polynomial of specified degree to small sections of the data and then computing the derivative of the polynomial over that section. The degree of the fitting polynomial is also requested in the prompt list.

Integral

Computes the running integral of the current segment.

Smooth

Smooths the current segment by convolution with a filter response computed from a lowpass Blackman window frequency response. Selection of a cutoff frequency (cycles per point) is available to allow flexibility in the amount of smoothing.

Pwr spec

Calculates the power spectrum of the current segment. (The power spectrum is the square magnitude of the Fourier transform, FFT.) The result of the FFT calculation is linearly compressed so that it has the same number of points as were in the current segment. This compression is done to facilitate the comparison of frequencies found in waveforms containing different numbers of points.

The routine also finds the minimum and maximum frequencies of the power spectrum. The frequency limits are printed on the screen in **Index** and **World** units as cycles/wfrm and cycles/(horizontal unit), respectively. These numbers may be saved in the current note file by executing Note #'s in the **Analysis** menu.

Envelope

Finds the envelope of the current segment by using the Hilbert transform techniques.

6.7 Analyzing the Current Segment

The **Analysis** menu lets the user extract quantitative information about waveform features. Basic statistics, rise time, peak width, etc. may be calculated and displayed. The user may then print these results or store them in a note file. Analysis functions are inherently descriptive, and do not alter the contents of any waveform repository.

If plotting is on, selecting each Analysis option causes SEG to be replotted and two graphics cursors (vertical lines) to appear on the plot. These cursors can be used to locate an arbitrary region of the segment for analysis. The **Ins** key initiates calculation over the circumscribed area and displays the results. If plotting is off, cursors are not made available, and analysis is performed on the entire SEG waveform. A complete description of the numeric pad function keys used to control the graphics cursors follows:

- Left arrow** -- Moves either or both cursors to the left.
- Right arrow** -- Moves either or both cursors to the right.
- End** -- Makes the left cursor active; the right cursor is disabled.
- PgDn** -- Makes the right cursor active; the left cursor is disabled.
- Down arrow** -- Enables both left and right cursors.
- PgUp** -- This key is used to set the distance increment which the cursors will move on each strike of the left or right arrow keys. A number is expected after the **PgUp** key is struck; the larger the number, the greater the increment.
- Home** -- Displays the current horizontal and vertical coordinates of the data array at the active cursor.
- Ins** -- Initiates analysis on the region of the current segment between the left and right cursors.

Each analysis operation prints results in the lower right corner of the screen. While still on the screen, they can be output to the current note file by executing **Note #**'s. Any computation dealing with the horizontal axis references the **Index/World** status to determine which units to use.

The Analysis menu options are:

Rise time

Computes the horizontal span between the points where the waveform rises from a 10% amplitude level to a 90% level. The 10% and 90% levels are computed by normalizing the vertical range between the minimum and maximum amplitudes found between the left and right cursor marks. Lines are drawn on the plot to indicate the 10% and 90% levels and the horizontal region associated with the rise.

Fall time

Computes the horizontal span between the points where the waveform falls from the 90% amplitude level to the 10% level. The 10% and 90% levels are computed by normalizing the vertical range between the minimum and maximum amplitudes found between the left and right cursor marks. Lines are drawn on the plot to indicate the 10% and 90% levels and the horizontal region associated with the fall.

Stats

Computes basic statistics of the current segment region. Lines are drawn on the plot to indicate the mean and standard deviation bounds (i.e., mean +/- one standard deviation).

Curve area

Computes the area under the current segment region. The routine does more than compute a numerical sum of data points: it uses the horizontal increment between data points to produce the true curve area in real-world units or in points. For example, if two waveforms are identical, with the exception that the first spans a horizontal range twice as large as the second, then the first waveform will also have an area twice as large as the second.

Peak width

Computes the width of a peak between the left and right cursors. The user specifies the level at which the width is to be computed (the width-level) as a percentage of the total peak height. The peak height is calculated from the vertical separation between the maximum amplitude of the peak and the minimum of either the left or right cursors. Thus, this routine gives local peak width, not one relative to zero amplitude. Lines are drawn on the plot to indicate the width-level and the width. For proper computation, both the left and right cursors should be below the width-level.

Note #'s

Outputs the numbers in the lower right corner to the current note file or the printer. An appropriate header is provided. For example, with left and right cursor locations of 245 and 457 respectively, numbers produced by the "Fall time" option will be preceded by:

```
Fall time
Lf mark: 245 Rt mark: 457
```

6.8 Memory Operations

Two options are available for performing operations with the MEM segment. The first uses the **Memory Ops** menu to perform simple operations between the current and memory segments. These are described below. The second option uses the Mini Calculator to operate on the data in MEM. Once the Mini Calculator is opened, the memory segment may be pushed on the stack by typing **MEM** on the command line. Conversely, a one-dimensional, non-complex data array of a size identical to the current contents of the memory segment repository may be placed into the memory segment with the command word **>MEM**.

The **Memory Ops** options are:

Seg + Mem

Opens a prompt list for specifying a scale factor used to multiply SEG (SFS) and a scale factor used to multiply MEM (SFM). The computation is:

$$[SFS * (SEG)] + [SFM * (MEM)]$$

The result is stored in SEG. PRV is updated with the prior contents of SEG.

Seg - Mem

Opens a prompt list for specifying a scale factor used to multiply SEG (SFS) and a scale factor used to multiply MEM (SFM). The computation is:

$$[SFS * (SEG)] - [SFM * (MEM)]$$

The result is stored in SEG. PRV is updated with the prior contents of SEG.

Seg * Mem

Opens a prompt list for specifying a scale factor used to multiply SEG (SFS) and a scale factor used to multiply MEM (SFM). The computation is:

$$[SFS * (SEG)] * [SFM * (MEM)]$$

The result is stored in SEG. PRV is updated with the prior contents of SEG.

Seg / Mem

Opens a prompt list for specifying a scale factor used to multiply SEG (SFS) and a scale factor used to multiply MEM (SFM). The computation is:

$$[SFS * (SEG)] / [SFM * (MEM)]$$

The result is stored in SEG. PRV is updated with the prior contents of SEG.

Store

Stores SEG in MEM, overwriting the old memory data.

Exchange

Exchanges the contents of SEG and MEM.

CHAPTER 7: **FILE INPUT/OUTPUT**

Stack Contents	Parameters
(empty stack)	A= 1.0000 B= 5.0000 C= .2000 D= 1.0000 E= 2.0000 F= .5000 G= .0000 H= .0000 I= .0000
File I/O	Variables
File/Var Transfer Convert File Types Return	EKGdata R= 400 REAL BASIC S= 200 REAL 1-2-3 T= 200 REAL U= .0000 V= .0000 W= .0000 X= .0000 Y= .0000 Z= .0000

SCREEN R7.1

7.1 Overview

7.1.1 GENERAL DESCRIPTION

File I/O offers two forms of data transfer. The **File/Var Transfer** option moves data between ASYSTANT disk files and Desk Calculator Variables. The **Convert File Type** option converts data between DIF and ASCII file formats and the ASYSTANT file format.

7.1.2 DATA TRANSFER BETWEEN FILES AND VARIABLES

File/Var Transfer provides a number of options for working with ASYSTANT files. Data in an ASYSTANT file may be read into a Variable. Or

a Variable array may be written out to a new or existing data file. Comments may be entered and edited in file headers to simplify "housekeeping" of files.

ASYSTANT data is stored on disk in "subfiles". Each subfile is essentially an array on disk, having data type, precision, and dimension. The data read from or written to a disk file can span one or all rows of the subfile and a specified number of columns located anywhere within the subfile. If two or more subfiles of identical size are contiguous, then data I/O can span subfile boundaries.

In writing to an existing subfile, output data should be of the same numeric type, precision, and shape as the data on disk. A new subfile can also be created by appending a Variable array to a file, in which case ASYSTANT automatically uses the type, precision, and dimension of the Variable array.

Detailed information on the ASYSTANT file structure and data transfer can be found in Sections 7.2 and 7.3.

7.1.3 FILE TYPE CONVERSION

Convert File Types lets data files created by the ASYSTANT be converted into two popular file formats: DIF and ASCII. Files created in these formats by other programs, such as Lotus 1-2-3 or BASIC, can also be converted to the ASYSTANT file format.

DIF format is supported by Lotus 1-2-3 and many other spreadsheets. It is essentially an ASCII file with a header, which is organized as a single one- or two-dimensional array. Thus, each ASYSTANT subfile must be converted into a separate DIF file.

ASCII files are more flexible than DIF files, but also require more effort to create. An ASCII file may contain one- or two-dimensional data or data that is not formatted in any regular way. It may also contain multiple sets of data of differing shape and format; eg., a 100 point waveform can be appended to a 5 by 5 matrix of data. This flexibility allows more options to be accommodated. For example, a sequence of ASYSTANT subfiles can be converted into multiple ASCII "subfiles". If an ASCII file is to be converted, a range can be specified, beginning on a selected entry. ASYSTANT accommodates ASCII files with both unformatted and formatted data structure.

Additional details can be found in Section 7.4.

7.2 ASYSTANT Data File Structure

7.2.1 GENERAL DESCRIPTION

All ASYSTANT data files are constructed with the following format: Comments, Subfile 1, Subfile 2, ..., Subfile N.

Comments are composed of character strings. Data file comments can contain as many as 64 Kbytes of characters.

Each subfile contains either a one- or two-dimensional array. As with arrays, each subfile has associated with it a data type (integer, real, or complex) and precision (single or double). Data points are referred to by analogy as an array. If the subfile is one-dimensional, data is referred to by column number: the 10th data point is in the 10th column. For two-dimensional subfiles, the data is located by row and column: the data point in the 3rd row and 15th column has indices (3 , 15) in the subfile. Subfiles must each be smaller than 64K bytes in size; the maximum number of data points allowed depends on the precision and data type of the subfile. See Table 2.1 in Chapter 2 for details.

7.2.2 SUBFILE BLOCKS

For many applications, each subfile can be considered as a separate data array on disk. However, for some applications, most notably data acquisition, it may be necessary to write out continuous data spanning many subfiles.

ASYSTANT lets you interpret subfiles as being continuous by letting you select datasets that span contiguous subfiles within a block. A block is defined as a sequence of contiguous subfiles with identical size and shape. Figure 7.1 illustrates a typical data file. This file contains four blocks: subfiles 1 through 3; 4 through 8; 9 through 11; and subfile 12 by itself. The first three subfiles might be separate arrays computed in a simulation, subfiles 4 through 8 might be the continuous output of a 5 channel, strip-chart experiment conducted with the ASYSTANT+, and subfiles 9 to 12 might be the results of processing sections of the experimental data. Note that the first 8 subfiles and subfile 12 are two-dimensional, while subfiles 9 to 11 are one-dimensional.

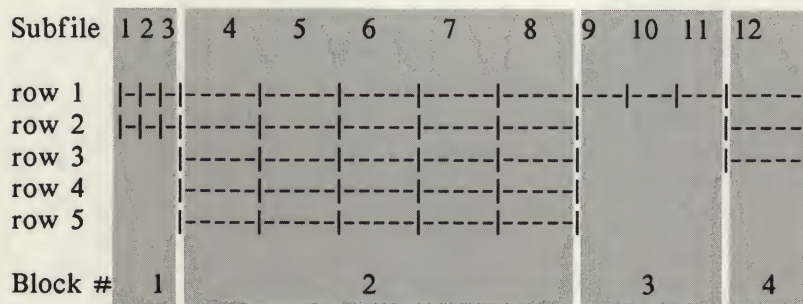


FIG.R7.1 - Schematic of subfile structure within a data file. Each "-" implies 10 data points, and "|" represents a subfile boundary.

7.2.3 DEFINITION OF A DATASET

A dataset describes an array of data, which may reside in a portion of a single subfile or span many subfiles. A dataset may not, however, cross a block boundary (i.e., span two subfiles of different shape). A dataset is located by its start column and its length in columns. As long as a block boundary is not violated, ASYSTANT will draw on as many subfiles as necessary to create the dataset for the specified location and size. A dataset may span all rows or only one row of a two-dimensional block. A dataset must be less than 64K bytes in size.

7.2.4 SELECTION

A dataset is chosen by specifying a subfile, row (if data is two-dimensional), start column, and number of columns. The specified subfile is used as the origin of the start column. For example, **Start column of 10, Subfile 5**, means the tenth column counting from the beginning of subfile 5. The start column can exceed the size of the specified subfile, so that the initial column of data resides in another subfile. However, the start column cannot extend beyond a block boundary. If the subfile is two-dimensional, either a particular row or all rows may be specified. If the subfile is one-dimensional, the row specification is superfluous; the first row will always be used. Finally, the number of columns completes the size of the dataset; eg., with a start column of 10 and the number of columns being 5, the dataset will contain columns 10, 11, ..., and 15. In Figure 7.2, the specifications are **Subfile 4, Start column 21, Row 3, Number of columns 150**.

Subfile	1	2	3	4	5	6	7	8	9	10	11	12
row 1												
row 2												
row 3												
row 4												
row 5												
Block #	1	2	3	4								

FIG.R7.2 - Location of a dataset within a data file. Each "-" implies 10 data points. The "*" denotes the location of the dataset with Parameters of subfile 4, row 3, start column 21, and number of columns 150. "|" represents a subfile boundary.

The dataset can also be selected graphically by scrolling through the data. Scrolling is restricted to a block (i.e., contiguous subfiles with identical shape and size). The specified start column is used as the initial scrolling column. The scroller will automatically adjust the specified subfile to start at the beginning of the block of similar subfiles. In the above example, a specified subfile and start column of 6 and 21, respectively, will be changed by the scroller to subfile 4 and start column 121. This shift is done to allow you to scroll through an entire block of subfiles. The number of columns is not used by the scroller; it tries to grab as much data as possible from the file, up to 1280 columns at a time.

TIP

Because a block of data may be very large, the scroller scales its display using only data surrounding the specified start column. If the dynamic range varies significantly in other parts of the signal, you may find the waveform is clipped (if the signal has a higher dynamic range) or is too compressed (lower dynamic range). If a particular region of the waveform is not well displayed, restart the scroller at the new position. This will cause the display to be rescaled appropriate to the current signal levels.

You can graphically select a dataset with the Scroller's left and right cursors. If cursors are not used, the dataset will default to the data window displayed on the screen.

Small blocks that can be contained completely on the screen are plotted with axes. Left and right cursors are provided to mark the dataset.

7.2.5 DATA TRANSFER

Datasets may be selectively read in from a file to calculator Variables or data may be output from a Variable to a file. Data output can either be written or appended.

Appending data is simpler than writing. Data in a Variable is added to the end of the file as a single subfile. If the file is initially empty (eg., a new file) **Append** must be used. The appended subfile becomes the first subfile. **Append** automatically creates the new subfile for the size and data type of the source Variable.

Data written into a dataset overwrites the existing data at that location. It is used to replace data which is no longer needed (eg., a waveform at an intermediate processing stage which is now superseded by later processing).

Consequently, beware that writing data requires that the Variable match the data type, precision, and size of the dataset on disk. Problems or unexpected results will arise if

1. the magnitude of a written number exceeds the precision of the dataset number type,
2. if the data type increases in hierarchy, i.e. from integer to real to complex, or
3. if the written array is a different shape than the dataset.

Only the first and last items will cause an error, so be aware that *some processing steps can alter the type, precision, and shape of a processed dataset compared to the input dataset.*

7.2.6 EDITING DATA FILE COMMENTS

Text may be entered in comment lines after a file has been opened for data transfer. A simple text editor is used for this task: text may be deleted, inserted, and overwritten. New data files created by the ASYSTANT are setup with 8 (initially blank) comment lines, each 64 characters long.

When the comments are not being edited, only the leftmost 40 characters of each line are visible. Moreover, if the file contains more than 8 comment lines, only the first 8 are visible. Upon entering the comment editor, all 64 characters become visible, and contiguous blocks of 8 lines become available for editing with the **PgUp** and **PgDn** keys.

7.3 File/Var Transfer

File Input/Output			
File Name: <C:\ekgdm0.dat>		Quit/Continue: * C *	
8 Comments		Subfiles (Total #: 10)	
1) EKG data / Acq rate 180 Hz 2) 20 mg Etorpine / Intravenously 3) 15:34 1 3/16 4) 5) Patient #: A34-6245 6) 7) 8)		Start#	Shape
		1	1 x 400
			#Repts
			10
I/O Variable (R - Z): < R > 4 x 4 INT			
Subfile: < 1 > 1 x 400 (Max Length: 4000)			
Row (0=all): < 0 >			
Start Column: < 1 >			
# of Columns: < 400 >			
Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit: * Q *			

SCREEN R7.2

7.3.1 GENERAL DESCRIPTION

File/Var Transfer performs data input/output between Desk Calculator Variables and ASYSTANT data files on disk. The following prompts are displayed in order of their appearance.

File Name

Selects the current data file and gives a short listing of the comments and subfile sizes.

Any string entered into this prompt is interpreted as a file name. This name should be structured with the usual DOS syntax of: d:filename.ext (drive: filename.extension). If you incorrectly specify a file name, press the **backspace** key to delete previous characters.

If the selected data file is not located on the disk, ASYSTANT assumes you want to open a new file and lists eight blank comment lines and no subfiles. The new file is not actually created until **Continue** is selected (see below), so you don't have to worry about filling your disk with files having misspelled names.

We recommend that you always explicitly specify a drive and file extension. If the drive, filename, or extension is not entered, the one last specified (displayed in the default prompt) will be used. On initial entry, the default drive is the drive that was selected for Temporary Work Files during setup and the default filename is NONE, with no extension.

TIP

To avoid defaulting to the last file extension used (i.e., to specify a null extension), terminate the filename with a period (.), eg.:

MYFILE.

No extension will be appended.

Quit/Continue

Q returns to the File I/O menu. If a new file was specified it will *not* be created. **C** causes the following prompts to be displayed. If a new file was specified, it is now created on disk.

I/O Variable (R - Z):

Identifies the Variable to be used for data transfer between the Desk Calculator and a disk file. When a Variable is selected, its current contents are displayed to the right of this prompt.

Subfile

Specifies the subfile to be used as the data origin in data transfer.

Row (0=all)

Sets the row number in the subfile. If zero is entered, all rows will be used.

Start Column

Defines the first column to be transferred. This column is relative to the specified subfile. With 100 column subfiles, subfile 3, column 10 is the same as subfile 2, column 110.

of Columns

Specifies the size of the dataset in columns. With a start column of 21, entering a 5 for this prompt implies that columns 21, 22, ..., and 25 will be used in data transfer. The dataset must lie within a block of subfiles of similar size and shape.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Reads the specified dataset into the specified Desk Calculator Variable.

WARNING

The current contents of the chosen Variable will be destroyed when **Read** is selected. You may return to the Variable I/O prompt by pressing the Up arrow key to select a different Variable, if you wish, before reading the data.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Writes the contents of the specified Desk Calculator Variable into the specified dataset. The data is still present in the Variable after writing.

Beware that writing data forces the written numbers to match the data type, precision, and size as the dataset on disk. Problems or unexpected results will arise if:

1. the size of an written number is too large for the dataset number precision
2. if the data type increases in hierarchy, i.e. from integer to real to complex.
3. if the written array is a different shape than the dataset.

Only the first and last items will cause an error, so be aware that *some processing steps can alter the type, precision, and shape of a processed dataset compared to the input dataset.*

WARNING

The **Write** option overwrites the data currently in the specified dataset on file. Select **Append** (see below) if you wish to save the current contents. Data which is located outside of the dataset range will not be affected by **Write**.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Appends the contents of the specified Desk Calculator Variable onto the end of the current data file. A single new subfile is created. The data is still present in the Variable after appending.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Starts scrolling through the specified row within the current block of subfiles. The value for the start column is used to orient the left side of the scroller display window within the block of data. If not enough data points are available for scrolling, the data will be plotted with axes. In both cases, left and right cursors are available for marking the precise portion of the data that you wish to grab. Furthermore, the data near the start column is used to choose a scale for the graphical display. If the amplitude of the data varies considerably, it may be desirable to restart the scroller if the data amplitude becomes too small or large.

Upon exiting the scroller or data plot, the dataset subfile, start column, and # columns are set up to reflect the portion of the block of subfiles that was either displayed on the screen or marked with the left and right cursors.

If enough data points are available for scrolling, then the numeric pad keys will be setup as "scroller and marker" keys. Chapter 6, section 6.3, describes the use of these keys.

If there are not enough points to scroll, the following numeric pad keys will be installed instead:

- Left arrow** -- Moves either or both cursors to the left.
- Right arrow** -- Moves either or both cursors to the right.
- End** -- Makes the left cursor active; the right cursor is disabled.
- Pg Dn** -- Makes the right cursor active; the left cursor is disabled.
- Down arrow** -- Enables both left and right cursors.
- Pg Up** -- This key is used to set the distance increment which the cursors will move on each strike of the left or right arrow keys. A number is expected after the **Pg Up** key is struck; the larger the number, the greater the increment.
- Home** -- Displays the current horizontal and vertical coordinates of the data array at the active cursor.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Plots the specified row of the currently specified dataset. If all rows are specified (with 0 = all), then only the first row will be plotted.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Opens up the screen for editing comments and installs the numeric pad keys with text editor features. The **PgUp** and **PgDn** keys allow more than 8 comments to be edited, if available.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Clears the screen and performs a complete list of the subfile blocks for the current data file. This feature was included since the normal **Subfiles** window only allows a listing of 7 different blocks of subfiles, and some files may have more than this number of blocks. On escaping from the listing, the screen will be restored.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Clears the screen and restarts the file specification procedure. If a new file was just specified, the file will be deleted unless data was appended or the comments were edited.

Read/Write/Append/Scroll/Plot/Edit/List/New file/Quit

Returns to the **File I/O** menu. Filename, I/O Variable, and dataset location are saved for the next use of the **File/Var Transfer** prompt list.

7.4 Converting File Types

Stack Contents <empty stack>	Parameters A= 1.0000 B= 5.0000 C= .2000 D= 1.0000 E= 2.0000 F= .5000 G= .0000 H= .0000 I= .0000
File Conversion Options DIF) Asystant Asystant) DIF ASCII) Asystant Asystant) ASCII Return	Variables EKGdata R= 400 REAL BASIC S= 200 REAL 1-2-3 T= 200 REAL U= .0000 V= .0000 W= .0000 X= .0000 Y= .0000 Z= .0000

SCREEN R7.3

7.4.1 GENERAL DESCRIPTION

Most software on the market uses a file format tailored to its unique needs in order to obtain maximum performance. ASYSTANT is no exception: it shares its file format with its sister product -- ASYST -- and no other commercial product.

However, a few standard data file formats do exist which can be used by most major programs and languages. ASYSTANT is capable of converting between two of the most popular formats, ASCII and DIF, which can be read by BASIC and many spreadsheet programs.

The **Convert File Types** option opens a series of menus designed to let you specify conversion of DIF and ASCII files into ASYSTANT format and vice versa.

7.4.2 TEXT IN ASCII AND DIF FILES

Both ASCII and DIF files may contain text interspersed with numbers. However, such mixing is not allowed in ASYSTANT files. Text found during conversion of an ASCII or DIF file is discarded. The deletion will not affect any other portion of the file; all numbers that are not designated as text will be

converted. Please note that text in an ASCII file is delineated by two surrounding double quotation marks (eg. "TEXT"), while text in a DIF file is denoted by a data type indicator.

7.4.3 CONVERTING ASCII FILES TO ASYSTANT

ASCII data can be stored on disk in almost any format. However, to make automatic conversion from ASCII to ASYSTANT files possible, a few rules are needed. ASYSTANT can read any ASCII file consistent with the IBM BASIC standard: numbers must be delimited by a comma or a space, and text must be contained within double quotation marks (eg. "TEXT"). If these rules were followed, ASYSTANT should be able to read a previously created ASCII file.

Even if the individual numbers can be read, the flexible organization of the ASCII file creates difficulties. Specifically, ASYSTANT organizes data in rectangular arrays, while an ASCII file does not. Furthermore, even if the ASCII file is meant to be structured as a rectangular array -- with one row per line -- the varying number of characters in each number might cause a row to spill over into a subsequent line. These nonuniformities could cause unexpected results in the output if the data were blindly stuffed into a rectangular array.

Consequently, ASCII files can be converted into ASYSTANT files by two methods:

1. an *unformatted* conversion which places all numbers found on a single line of text into a single row in the output ASYSTANT subfile, and
2. a *formatted* conversion which places a specified quantity of numbers into each row of the output ASYSTANT subfile; the numbers may span several lines of text.

Both methods let you specify the character used to delimit successive numbers (usually a comma). You also specify the number type (eg., integer) and precision (eg., double) of the subfile into which the converted data will be placed. The numbers in the ASCII file can be real or integer, in floating point or exponential format, and single or double precision. Both methods also allow you to specify an offset into the file at which you would like to begin the conversion.

For example, suppose you have a header at the beginning of the ASCII file that is 10 numbers long and contains information about the experimental setup; the actual data follows this header. You can begin conversion on the 11th number, skipping over the header and converting only the data. As another example, suppose you have two runs of 1000 data points both stored in a single ASCII file, you can convert only the second run by selecting an offset of 1001.

* **Unformatted conversion** allows you to specify the number of lines to convert, with 0 interpreted as all lines. For example, suppose your ASCII file has a header of 10 numbers and data from two experiments each occupying 15 lines. You can read the first experiment by selecting offset 11 and number of rows 15. The number of columns in the output subfile will be set automatically by the number count of the longest single line in the ASCII file. Shorter ASCII lines will be padded with zeros to fill out the subfile rows. (Remember, the numbers per line may differ in the ASCII file, but the ASYSTANT subfile must be rectangular).

As a specific example, suppose the following ASCII file is to be converted into a single precision, real subarray. You have chosen to convert 3 rows starting at entry 5.

```
-15.0 , 21.5 , 512 , 8.25E6      HEADER FOR FILE #1
10.1 , 20.7 , 165.32
1673. , -164 , 3.456E5 , 2.75E6   DATA FOR RUN 1
124.43 , 9465.1
986.02 , 23.9 , -36.467
9376.8 , -146.7 , 2.497E5 , 3.00E6 DATA FOR RUN 2
175.5 , 8452.72
```

The output subfile will appear as:

```
10.1      20.7      165.32      0.0
1673.     -164.0     3.456E5     2.75E6
124.43    9465.1     0.0        0.0
```

* **Formatted conversion** lets you to specify the shape of the output subfile independent of the organization of the ASCII data. The output subfile can be made either one- or two-dimensional, to accomodate one-dimensional waveforms or two-dimensional arrays.

A number of prompts simplify conversion of formatted data. If zero is specified for the number of columns, a one-dimensional output will result. ASYSTANT will simply read numbers one at a time, and append them to the end of the subfile. If you enter a value other than 0 for the number of columns, a second prompt will appear, letting you specify the number of rows. A zero for number of rows means that ASYSTANT will read the entire ASCII file, one number at a time, adding to the output subfile row by row as needed. The organization of the ASCII file does not affect the contents of the subfile: if ASYSTANT reaches the end of a line before it fills a subfile row, it will continue with the first number on the next line. Of course, you may enter a specific number of rows in the prompt: conversion will stop when the subfile is filled.

For example, suppose formatted conversion were selected for the ASCII data above, starting on entry 5, number of columns 3, and number of rows 3. The output subfile will be structured as:

10.1	20.7	165.32
1673.	-164.0	3.456E5
2.75E6	124.43	9465.1

Note that the fourth number on the third row of text (i.e., 2.75E6) became the first entry on the third row of the subfile: no 0's have been padded as happened in unformatted conversion. However, if the total quantity of numbers in the ASCII file, from the offset to the end, is not a multiple of the chosen number of columns, then the last row will contain enough zeros to fill out the end of the row.

7.4.4 CONVERTING DIF FILES TO ASYSTANT

The DIF file format is much more clearly defined than ASCII. DIF files are always rectangular -- like ASYSTANT subfiles -- because of the structures imposed by spreadsheet programs, where DIF files originated.

A detailed explanation is not necessary for conversion of DIF files to ASYSTANT files. It is sufficient to note that any text in a DIF file will be ignored in the conversion to ASYSTANT format. For example, if the source of the DIF file is a spreadsheet program and the spreadsheet contained a column (or row) of text, then the ASYSTANT subfile will have one less column (or row) than the original spreadsheet. The numbers in the DIF file can be real or integer, in floating point or exponential format, and of single or double precision. You may choose the type (eg., integer) and precision (eg., double) of the output subfile.

Note that, although spreadsheet programs use their own file formats internally, these programs -- including Lotus 1-2-3 -- supply file conversion routines that enable you to convert your spreadsheet into DIF format and vice versa.

7.4.5 CONVERTING ASYSTANT FILES TO DIF AND ASCII

ASYSTANT subfiles may also be converted to DIF or ASCII files, to be used with other programs. The conversion routine allows you to select the field widths and notation of the output numbers. For example, you may output the numbers as dollars and cents by selecting a total field width of 10 for the entire number, a field width of 2 for the number of decimal places, and fixed point notation. Likewise, you could output numbers in scientific notation with a variable total field width and 3 decimal places. In the case of ASCII files, you can also select the delimiter used to separate successive numbers; a comma is recommended.

If the ASYSTANT file contains multiple subfiles, you may select the particular subfile or subfiles that you wish to convert. Since DIF files can only accomodate one "spreadsheet", you can convert only one ASYSTANT subfile per DIF file. However, since ASCII file formats can accommodate multiple "subfiles", the conversion routine allows you to convert a range of subfiles from the ASYSTANT file.

7.4.6 MENU ITEMS

File conversion options are listed below:

DIF > ASYSTANT

Opens a prompt list allowing filename specification of the input DIF file (to be converted) and the output ASYSTANT file. Other prompts specify the type and precision of the output subfile. **Quit/Continue** lets you abort or continue with the conversion. If the selected output file does not exist, it will be created and given 8 comments. In any case, the output subfile is appended to the end of the file. (The comments can be edited within the **File/Var Transfer** option.)

ASYSTANT > DIF

Opens a prompt list allowing filename specification of an input ASYSTANT file and an output DIF file. If the ASYSTANT file has more than one subfile, you must select a particular subfile for conversion (default is subfile 1). **Quit/Continue** lets you abort or continue with the conversion. If you continue, a second prompt list allows you to select the field width and notation (fixed point or scientific) of the output numbers. A second **Quit/Continue** prompt is provided. Continue performs the conversion.

ASCII > ASYSTANT

Opens a prompt list allowing filename specification of the input ASCII file and the output ASYSTANT file. Additional prompts select the type and precision of the output subfile.

Quit/Continue lets you abort or continue with conversion. If you elect to continue, a second prompt list of conversion parameters is opened (see below). Responding **Continue** in this second list begins conversion. If the selected output file does not exist, it will be created and given 8 comments. In any case, the output subfile is appended to the end of the file. (The comments can be edited within the **File/Var Transfer** option.)

The second prompt list contains:

Formatted/Unformatted

Allows you to select between a formatted or an unformatted conversion. The formatted conversion allows the number of rows and columns in the output subfile to be specified. The unformatted conversion takes the numbers on each line of text and places them in a single row of the output subfile; a rectangular subfile shape is maintained by padding short rows with zeros.

Start conversion on entry

Conversion will begin on the selected entry. Eg., if the conversion should skip the first 9 numbers in the file, enter 10 at this prompt.

Number of cols (0=all)

Designates the number of columns that the output subfile should contain. This prompt appears only with formatted conversions. If zero is chosen, then the output subfile will be one-dimensional and the conversion process will continue until no more data is found.

Number of rows (0=all)

Designates the number of rows that the output subfile should contain. This prompt appears only with formatted conversions. If zero is chosen, then the conversion process will continue until no more data is found; the final row may contain some zeros to fill the remaining columns if not enough numbers are found. If the number one is entered, the output will be one-dimensional. This prompt does not appear if zero columns have been selected.

Delimiter character

Allows selection of the character used to separate numbers in the file. Generally, this character is a comma or a space.

Quit/Continue

If **Quit** is chosen, you are returned to the File Conversion menu; **Continue** begins the conversion.

ASYSTANT > ASCII

Opens a prompt list allowing specification of the input ASYSTANT file and the output ASCII file. If the input ASYSTANT file has more than one subfile, two prompts are made available for selecting the range of subfiles to be converted. **Quit/Continue** lets you abort or continue with the conversion process. If you continue, a second prompt list opens for selecting the field width and notation (fixed point or scientific) of the output numbers. You may also specify the delimiter character used to separate the output numbers. **Continue** in the second prompt list causes the conversion to start.

CHAPTER 8: **FILE PROCESSOR**

Dataset Location		I/O Status	
Start subfile:	5	Filename	# Subfiles
Row # (0 = all):	0	Input: C:EKGDMO.DAT	10
Start column:	200	Output: C:EKGDMO.DAT	13
# columns:	600	Operation: DS.Cat.&.Smooth	Unary
Offset (columns):	600	Input var(s): R	
# datasets:	3	Output var: T	Append
		Current DS: 3	

Main Menu	Variables
File > File	R= 600 REAL
File > Var	S= .0000
Var > File	T= 600 REAL
Return	U= .0000
	V= .0000
	W= .0000
	X= .0000
	Y= .0000
	Z= .0000

SCREEN R8.1

8.1 Overview

8.1.1 GENERAL DESCRIPTION

The File Processor (**File Proc**) provides a comprehensive set of processing procedures that allows you to automatically process large and/or multiple datasets. While processing of disk-based data can be done manually in steps by reading data into a Calculator, processing it, and then returning it to disk, File Proc performs these steps automatically.

Any Desk Calculator function may be applied to file data using File Proc. For example, a waveform may be smoothed and then integrated, two-dimensional FFT and IFFTs applied to square arrays, and autocorrelations performed. For ease of use, the most commonly used Desk Calculator functions are provided in a File Processor menu. However, additional and multi-stage processing steps can be built up with User Functions (see Chapter 3), providing virtually unlimited file-processing capabilities.

In addition to the functions available in the Desk Calculator, some procedures specific to file-processing have been incorporated into the File Processor. These procedures include smoothing across waveforms larger than 64K (as might be produced by the Strip Chart Recorder or Data Logger of the ASYSTANT+) and the computation of inter waveform statistics (as opposed to the intra waveform statistics provided with the Wave Proc and Stats menus).

8.1.2 PROCESSING DATA

The File Processor reads data from an input source, processes it internally, and transfers the results to an output source. The transfer of data can occur in three directions:

1. File to file (**File > File**). Data are both input from and output to disk file. The same or different files may be used for input and output. If different files are used, the input file is not affected by the operation.
2. File to Variable (**File > Var**). Data are input from disk file and output is stored in a Desk Calculator Variable. The disk file remains unchanged.
3. Variable to file (**Var > File**). The contents of a Calculator Variable are input and output is directed to a disk file. The Variable contents are not changed by the operation.

All three methods require Desk Calculator Variables to buffer input and output data during processing. For example, if **File > File** processing has been selected, each input dataset is first read from disk and placed in an input Variable. The input Variable contents are processed and then placed in an output Variable. The output Variable contents are then written to the output disk file.

Four steps are needed to set up a processing procedure and processing begins only after all four stages have been completed. These are:

- Step 1:* Specify the input and/or output data files.
- Step 2:* Specify the input dataset(s) for processing.
- Step 3:* Select the processing operation.
(Pre defined User Functions may be used).

Step 4: Select the input and output Variables to use as processing buffers.

File Proc operations are either unary or binary. Unary operations, such as **Sqrt** (square root) and **Abs** (absolute value), require only one input argument, while binary operations, such as **+** and **-**, require two. With binary operations, only one of the two input operations can be drawn automatically from a disk file; the second argument must be a Variable that already contains the appropriate data.

8.1.3 SCREEN LAYOUT

The File Processor's screen (refer to Screen R8.1) is continually updated to keep you informed of the current setup specifications and processing status.

The window titled **Dataset Location**, in the upper-left corner of the screen, list the location of the data to be processed within the input data array. Section 8.3 describes the prompts used in specifying the dataset location.

The window in the upper right, titled **I/O Status**, displays the input and/or output file names and the number of subfiles in each. Also listed are the most recently selected processing operation, the input and output Variables, and the method of writing to the file (**append** or **inplace**). During processing **Current DS** is updated to reflect the dataset number currently being processed. (Multiple datasets may be processed with File Proc -- see section 7.2 for the definition of datasets.)

In the lower-right window all of the Desk Calculator Variables and their contents are shown.

8.2 Considerations in Selecting and Processing Data

Essential to understanding the processing of a data file is an understanding of how ASYSTANT organizes data on disk and how data are transferred to and from Variables. Chapter 7, section 7.2, describes the ASYSTANT data file structure. If you have not already read this section, please do so before continuing.

8.2.1 PROCESSING DATASETS

As described in section 7.2, a dataset is chosen by specifying a subfile, row (if the dataset is two-dimensional), start column, and number of columns. The dataset may cross subfile boundaries but must lie wholly within the block of identically shaped subfiles.

File Proc can automatically process both single and multiple datasets within a block. The location of successive datasets is determined by the column offset setting. For example, an offset of 100 means that the next dataset will begin at column $(100 + S)$, where S is the start column of the previous dataset. By default, File Proc will try to process as many datasets as can be completely contained within the current block. You may, of course, choose to process less than the maximum number of datasets. Figure R8.1 illustrates the specification of multiple datasets which are not contiguous.

Subfile	1	2	3	4	5	6	7	8	9	10	11	12
row 1		-			-		-		-		-	
row 2		-			-		-		-		-	
row 3		-		1	1	1	2	2	2	3	3	3
row 4		-			-		-		-		-	
row 5		-			-		-		-		-	
Block #	1	2						3			4	

FIGURE R8.1 - Location of a dataset within a data file. Each - implies 10 data points. The successive 1's, 2's, and 3's denote the location of the first, second, and third datasets respectively. The first dataset starts in subfile 4 with row 3, start column 21, and 40 columns. The dataset offset is 60 columns and the number of datasets has been selected to be three.

Successive datasets will always have the same size and shape as the previous dataset. Note that a maximum of 4 datasets could fit in this block of subfiles, given the starting column.

Multiple dataset processing is not always a one-to-one transformation. Some File Proc operations, for example Dataset Average (DS Average), accept multiple datasets on input, to produce a single dataset on output.

8.2.2 DATA FLOW DURING PROCESSING

As Figure R8.2 illustrates, each original dataset flows from the *Input File* (if specified) to the Input Desk Calculator Variable *V1* and on through the *Operation*. The processed dataset then flows to the Output Variable *V3* and out to the *Output File* (if specified). For binary operations data already stored in a second input Variable, *V2*, is input as the second operand of the *Operation*. Multiple datasets are automatically processed in succession.

Thus, if both an input and an output file are specified (**File > File** processing), successive datasets are moved automatically from the input file, processed, and placed in the output file.

Input Specification

Input Variable *V1* must always be specified. If a Binary operation is selected, so must *V2*. **File > File** and **File > Var** processing require an input file to be specified, as well; successive datasets will be automatically moved into *V1* from the file during processing. If **Var > File** is selected, the contents of *V1* must be set up before processing since there is no input file. The contents of *V2* must always be setup prior to processing.

Output Specification

Output Variable *V3* must always be specified. If **File > File** or **Var > File** processing has been selected, an output file must also be specified. As each dataset is received by the output Variable *V3*, it is automatically moved into the output file. Only the last dataset is preserved in *V3* for any of the transfer options.

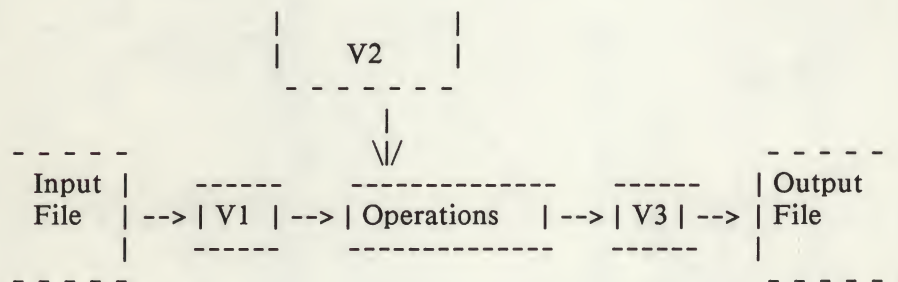


FIGURE R8.2 - Illustrates the flow of data during file processing. The Operations box accepts the input Variable(s) *V1* (and *V2* for binary operations), performs the operation, and presents the result to the output Variable *V3*. During processing, *V1* may automatically receive data from an input disk file and *V3* can, in turn, output data automatically to disk. *V2*, however, must contain the second binary operand before processing begins. Note that the Input File, Output File, and *V2* are not used in all cases.

8.2.3 GENERAL OUTLINE OF PROCESSING PROCEDURES

The three processing procedures, **File > File**, **File > Var**, and **Var > File**, are presented on entering File Proc. When one is selected, ASYSTANT presents sequentially the proper prompt lists for setting up the procedure. The specification prompts required for each type of processing are outlined below in the order of their appearance:

File > File

1. Input file
2. Output file
3. Processing operation
4. Input and output Variables

File > Var

1. Input file
2. Processing operation
3. Input and output Variables

Var > File

1. Output file
2. Processing operation
3. Input and output Variables

The next four sections describe each of the four steps in turn.

8.3 Input File Specification

The menu items **File > File** and **File > Var** each require an input file to be specified. The following prompts are used to specify an input file and locate the datasets to be processed.

Input File			
File Name: (C:EKGDMO.DAT)		Quit/Continue: * C *	
8 Comments		Subfiles (Total #: 10)	
1> EKG data / Acq rate 180 Hz	Start#	Shape	#Repts
2> 20 mg Etorpine / Intravenously	1	1 x 400	10
3> 15:34 I 3/16			
4>			
5> Patient #: A34-6245			
6>			
7>			
8>			
Subfile: (5) Row (0=all): (0) Start column: (200) 1 x 400 (Max Length: 2400) # of columns: (600) Offset (columns): (600) # datasets: (3) Scroll/Plot/Edit/List/New file/Quit/Continue: * C *			

SCREEN R8.2

File Name

Selects the current data file and gives a short listing of the comments and subfile sizes.

Any string entered into this prompt is interpreted as a file name. This name should be structured with the usual DOS syntax of d:filename.ext (drive: filename.extension). If you incorrectly specify a file name, press the **backspace** key to delete previous characters. If the selected data file is not located on the disk, File Proc will issue an error message.

We recommend that you always explicitly specify a drive and file extension. If the drive, filename, or extension is not entered, the last one specified (displayed in the default prompt) will be used. On initial entry, the default drive is the drive that was selected

for Temporary Work Files during setup and the default filename is NONE with no extension.

TIP

To avoid defaulting to the last extension used (i.e., to specify a null extension), terminate the filename with a period (.), e.g.:

MYFILE.

No extension will be appended.

Quit/Continue

Q returns to the File Proc menu. C causes the following prompts for dataset selection to appear:

Subfile

Specifies the subfile to be used as the data origin in data transfer.

Row (0=all)

Selects the starting row number in the subfile. If zero is entered, all rows will be used.

Start Column

Defines the first column to be transferred. This column is relative to the specified subfile. With 100-column subfiles, subfile 3 and column 10 is the same as subfile 2 and column 110.

of Columns

Specifies the size of the dataset in columns. With a start column of 21, entering a 5 for this prompt implies that columns 21, 22, ..., and 25 will be used in data transfer. The dataset must lie within a block of subfiles of similar size and shape.

Offset (columns)

Sets the position of a subsequent dataset relative to a previous one. With a start column of 21, an offset of 10 implies that the next dataset will have a start column of 31, the one after that will have a start column of 41, and so on. Note that this new start column is independent of the number of columns in the dataset, so that consecutive datasets may overlap if desired. By default, the offset will be set to the number of dataset columns. This defines a contiguous group of datasets.

Datasets

Specifies the number of datasets that you wish to process. The value is automatically set to the maximum number possible, given the settings of the other prompts. Only datasets that are completely contained within the block of subfiles can be processed. You should reset this prompt only if you wish to reduce the number of datasets processed.

Scroll/Plot/Edit/List/New file/Quit/Continue

Starts scrolling through the specified row within the current block of subfiles. The value for the start column is used to orient the left side of the scroller display window within the block of data. If not enough data points are available for scrolling, the data will be plotted with axes. In both cases, left and right cursors are available for marking the precise portion of the data that you wish to grab. Furthermore, the data near the start column is used to choose a scale for the graphical display. Where the amplitude of the data varies considerably, it may be desirable to restart the scroller if the data amplitude becomes too small or large.

Upon exiting the scroller, the dataset subfile, start column, and number of columns are updated to reflect the portion of the block of subfiles that was displayed on the screen or marked with the left and right cursors.

If enough data points are available for scrolling, then the numeric pad keys will be set up as "scroller and marker" keys. Chapter 6, section 6.3, describes the use of these keys.

If there are not enough points to scroll, the following numeric-pad keys will be installed instead:

- Left arrow** -- Moves either or both cursors to the left.
- Right arrow** -- Moves either or both cursors to the right.
- End** -- Makes the left cursor active; the right cursor is disabled.
- Pg Dn** -- Makes the right cursor active; the left cursor is disabled.
- Down arrow** -- Enables both left and right cursors.
- Pg Up** -- Sets the increment of distance that the cursors will move on each strike of the left or right arrow key. A number is expected after the **PgUp** key is struck: the larger the number, the greater the increment.

Home -- Displays the current horizontal and vertical coordinates of the data array at the active cursor.

TIP

Use **Scroll** to visually select a dataset for processing. Set up the dataset specifications' prompts for the general location of the data of interest. Then **scroll** through the array and mark precisely the endpoints of the subarray you wish to process. When you **Esc** back to the dataset specification menu, all of the numeric prompts will have been updated to match the dataset you defined graphically.

Scroll/Plot/Edit/List/New file/Quit/Continue

Plots the specified row of the currently specified dataset. If all rows were specified (with 0 = all), then only the first row is plotted.

Scroll/Plot/Edit/List/New file/Quit/Continue

Opens up the screen for editing comments and installs the numeric pad keys with text-editor features. The **PgUp** and **PgDn** keys allow more than 8 comments to be edited, if available.

Scroll/Plot/Edit/List/New file/Quit/Continue

Clears the screen and displays a complete list of the subfile blocks for the current data file. This feature was included since the normal **Subfiles** window only allows a listing of 7 blocks of subfiles and some data files may have more.

Scroll/Plot/Edit/List/New file/Quit/Continue

Clears the screen and restarts the file specification procedure.

Scroll/Plot/Edit/List/New file/Quit/Continue

Returns to the File Proc Main menu.

Scroll/Plot/Edit/List/New file/Quit/Continue

Accepts the input file and dataset specifications and proceeds to the next step.

8.4 Output File Specification

The menu items **File > File** and **Var > File** each requires an output file to be specified. The output file may be a new or an existing file and may be the same or different from the input file. Two methods of file output are available: **append** and **inplace**.

Append *must* be used if the output file is different from the input file and may be used if they are the same file. This method causes the dataset in the output Variable to be appended to the end of the output file as a single subfile. (In a new file, this will be the first subfile). When multiple datasets are processed, each one will be automatically appended to the previous one as they flow from the output Variable.

If the output and input files are the same, then the **inplace** writing option can be specified. This method grabs data from the input dataset, processes the data, and writes the output data back into the same locations within the file; i.e., the input data is overwritten by the output data. This method is useful when only a portion of the data file needs to be processed or when available disk space is too limited for separate input and output files. If the input and output files are not the same, the **Inplace/Append** option will not be offered.

WARNING

Make a backup copy of the input data file before processing with the **inplace** option! The original input datasets will be lost!

Output File			
Output file: (C:EKGDMO.DAT)		Quit/Continue: * C *	
8 Comments		Output file = Input file	
		Subfiles (Total #: 10)	
1) EKG data / Acq rate 180 Hz	Start#	Shape	#Repts
2) 20 mg Etorpine / Intravenously	1	1 x 400	10
3) 15:34 3/16			
4)			
5) Patient #: A34-6245			
6)			
7)			
8)			
Output method: Append			
Output data: Inplace/Append/Edit/List/New file/Quit/Continue: * C *			

Inplace/Append/Edit/List/New file/Quit/Continue

Sets the writing mode to **inplace**. The same data file must be used for both input and output and the output data will be written back to the same location from which it was read. Data that is not included in the datasets selected for processing will not be affected.

Beware that writing **inplace** will force output numbers to match the data type, precision, and dataset-array size of the input data. Problems or unexpected results will arise if:

1. the size of an output number is too large for the input number precision,
2. if the data type increases in hierarchy (i.e., from integer to real to complex) or
3. if the output array is a different shape than the input dataset array.

Only the first and last items will cause an error, so be aware that *some processing steps can alter the type and precision of an output dataset compared to the input dataset.*

Inplace/Append/Edit/List/New file/Quit/Continue

Sets the writing mode to **append** the data to the end of the output file. Each processed dataset is appended as a single subfile. **Append** will automatically be used if the output file is a new file or is different from the input file.

Inplace/Append/Edit/List/New file/Quit/Continue

Opens up the screen for editing comments and installs the numeric pad keys with text-editor features. The **PgUp** and **PgDn** keys allow for more than 8 comments to be edited, if available.

Inplace/Append/Edit/List/New file/Quit/Continue

Clears the screen and displays a complete list of the subfile blocks for the current data file. This feature was included since the normal **Subfiles** window only allows a listing of 7 blocks of subfiles and some data files may have more.

Inplace/Append/Edit/List/New file/Quit/Continue

Clears the screen and restarts the file specification procedure. If the current file is new (i.e., just created), it will be deleted unless comments have been edited.

Inplace/Append/Edit/List/New file/Quit/Continue

Returns to the File Proc Main menu. If the current file is new (i.e., just created), it will be deleted unless comments have been edited.

Inplace/Append/Edit/List/New file/Quit/Continue

Proceeds to the menus of processing operations.

8.5 Processing Operations

Processing operations are either unary or binary, meaning that they require one input Variable or two, respectively. In either case, one Variable (*V1* in Figure R8.2) can come from an input file; the second, if needed, must contain data before processing begins.

A few processing options are designed specifically for processing data files. These options are denoted by the prefix **DS** for **Dataset**. They may not be used for processing Desk Calculator Variables with **Var > File**.

Note that selection of the menu items in the Unary and Binary menus *does not start processing*. Some operations will prompt for additional information and all operations require specification of input and output Variables before processing starts.

8.5.1 USER FUNCTION OPERATIONS

The operations listed in the Unary and Binary Operations menus were chosen for their general utility. Many more items could have been included, but were not because of space limitations. To circumvent this restriction, the Unary menu was augmented with the **User function** option. Any sequence of Desk Calculator functions that you assigned to a function key via the **Functions** option in the Desk Calculator Main Menu may be used as a processing operation in File Proc. (See Chapter 3, User Functions.)

For example, you could define **F1** to be **2d:fft**, thus allowing you to take two-dimensional fast-Fourier transformations of your data file. You can also define multiple processing steps in one User function to save overall processing time. For example, suppose you wish to integrate your data twice and compute the crosscorrelation of this result with another waveform that is stored in the **T** Variable. The end result, which would normally require multiple processing steps, can be had by defining the function key **F2** as:

```
F2      1> integ integ
        2> T crosscorr
        3>
        4>
        5>
```

and then selecting **User function F2** as the processing operation.

NOTE

User functions should be written to expect the input Variable *V1* on the stack, and return the processed output to the stack. File Proc will automatically transfer this result to *V3* for output. Functions can draw on Variables as needed for additional arguments.

8.5.2 SPECIAL KEYS

The processing operations are separated into two menus: **Unary ops** and **Binary ops**. Use the **Tab** key to toggle between them.

Esc will halt processing only at the end of the current dataset. This ensures a clean interruption. Note that after pressing **Esc** there may be a delay before control is returned to the user until the current dataset is processed.

Dataset Location		I/O Status	
Start subfile:	5	Filename	# Subfiles
Row # (0 = all):	0	Input: C:EKGDMO.DAT	10
Start column:	200	Output: C:EKGDMO.DAT	10
# columns:	600	Operation: User Funct 1	Unary
Offset (columns):	600	Input var(s): R	
# datasets:	3	Output var: T	Append
		Current DS: 0	

Unary ops		Binary ops	
Abs	Sqrt	+	-
Log	10^	*	/
Poly Scale	User function	Max	Min
FFT	IFFT	Catenate	Laminate
Pwr spec	Autocorr	Convolve	Crosscorr
Smooth	DS Cat & Smooth	Quit	
DS Average	DS Stndrd Dev		
DS Subsection	DS Reshape		
Quit			

SCREEN R8.4

8.5.3 UNARY OPERATIONS

Abs

Takes the absolute value of each element in the input Variable.

Sqrt

Takes the square root of each element in the input Variable.

Log

Finds the log base 10 of each element in the input Variable.

10^

Computes 10 raised to the power of each element (z) in the input Variable: 10^z.

Poly Scale

Opens a prompt list that allows specification of the coefficients of up to a 5th-degree polynomial. Each element in the input Variable is scaled by this polynomial. For example, if the polynomial is a pure cubic, the output will be the cube of the input.

User function

Opens a prompt list that allows you to choose a User-defined Function for the processing operation. (The User functions are defined in the **Functions** menu in the Desk Calculator) The function should expect *V1* on the stack, and return its answer to the stack. File Proc will automatically supply successive *V1*'s and return the answers to *V3*.

TIP

With the exception of the special **DS** operators (see below) any of the File Proc operators are available in a User function. If you want to perform multiple-processing operations, you can do it in one pass by setting up the appropriate User function.

FFT

Takes the discrete Fourier transformation of the input Variable and returns a complex value. If the Variable contains less than 2^n columns, with n being an integer, the transformation will be applied to the first $2^{(n-1)}$ columns. For example, if the Variable contains 256 ($=2^8$) columns, then 256 columns will be transformed. However, if only 255 columns exist (less than 2^8), only the first 128 columns will be transformed ($128 = 2^7$). Multiple rows may be transformed simultaneously with this option.

IFFT

Takes the inverse discrete Fourier transformation of the input Variable. If the Variable contains less than 2^n columns, with n being an integer, the inverse transformation will be applied to the first $2^{(n-1)}$ columns. For example, if the Variable contains 256 ($=2^8$) columns, then 256 columns will be transformed. However, if only 255 columns exist (less than 2^8), only the first 128 columns will be transformed ($128 = 2^7$). Multiple rows may be transformed simultaneously with this option.

Pwr spec

Finds the power spectrum of the input Variable via the square magnitude of the FFT. Note that, since this operation uses the

FFT, the operation will only be applied to the first (and maximum) 2^n number of columns in the Variable, as described in FFT above. Multiple rows may be operated on simultaneously with this option.

Autocorr

Computes the autocorrelation function of the input Variable. If the input Variable has L columns, then the output array will have $2L-1$ columns. **Autocorr** cannot be performed if the **inplace** output method was selected.

Smooth

Smooths the input Variable by convolution with a filter response computed from a low-pass Blackman window frequency response. Selection of a cutoff frequency (cycles per point) is available in a prompt list to allow flexibility in the amount of smoothing. Multiple rows may be smoothed simultaneously.

TIP

The Blackman window low-pass filter used in these operations is also provided in the Wave Processor. If you are about to smooth a large file, you may wish to smooth one or more datasets manually to judge the appropriate cutoff frequency.

DS Cat & Smooth

Consecutive datasets in long data files (greater than 64K) are catenated and then smoothed by convolution with a filter computed from a low-pass Blackman window frequency response. Selection of a cutoff frequency (cycles per point) is available in a prompt list to allow flexibility in the amount of smoothing.

The method uses a variation of the overlap-and-save technique, so that if consecutive datasets are continuous, no smoothing edge effects will appear across the boundaries of the datasets--the complete output waveform will appear to have been operated on by the **Smooth** option above. Multiple rows may be smoothed simultaneously with this option.

Note: Only data files can be processed with this option.

DS Average

Computes the average of consecutive datasets, not across the dataset, but across columns in the dataset. The output will be a single dataset with the same number of columns as each of the input datasets. Regardless of the number of input datasets, only

one dataset will be output. For example, if the three input datasets are (1,2,3), (9,5,7), and (3,7,5), the output dataset will be (4.33,4.67,5.00). The datasets may have multiple rows.

Note: Only data files can be processed with this option.

DS Stndrd Dev

Computes the standard deviation of consecutive datasets, not across the dataset, but across columns in the dataset. The output will be a single dataset with the same number of columns as each of the input datasets. Regardless of the number of input datasets, only one dataset will be output. For example, if the three input datasets are (1,2,3), (9,5,7), and (3,7,5), the output dataset will be (4.16,2.52,2.00). The datasets may have multiple rows.

Note: Only data files can be processed with this option.

DS Subsection

Opens a prompt list for extracting a region of the input Variable. If the input Variable is one-dimensional, a start column, column increment, and the total number of columns to be extracted may be specified. If the input Variable is two-dimensional, then a start row, row increment, and the total number of rows to be extracted may also be specified. Note that the column specifications are relative to the beginning of each input Variable. Thus, regardless of the position of the dataset in the file (e.g., starting at row 1674), a start column of 1 in the subsection will always start the extraction at the first column of the dataset.

Note: Only data files can be processed with this option. **DS Subsection** is useful for creating a new data file containing only a specific subsection of each dataset.

DS Reshape

Allows adjustment of the number of rows and columns in the input Variable. A prompt list is used for selecting the number of rows and columns to be output. The total number of output points may be less than the input if either the number of output rows or the number of output columns does not divide evenly into the number of input points. For example, if the input Variable has 5 rows and 20 columns and the number of output rows is changed to 7, then the output Variable will have 7 rows and 14 columns and only the first 98 ($7 \times 14 = 98$) input points will be output.

Note: Only data files can be processed with this option.

Quit

Returns to the File Proc Main menu, without performing the processing operation.

8.5.4 BINARY OPERATIONS**+**

Adds the two input Variables as $V1 + V2$ on an element-by-element basis. If both input Variables are arrays, then they must both be the same size and shape. If one of the Variables is a number (usually the second input Variable), then the other (first) Variable can be a number or an array of any shape or size.

-

Subtracts the two input Variables as $V1 - V2$ on an element-by-element basis. If both input Variables are arrays, then they must both be the same size and shape. If one of the Variables is a number (usually the second input Variable), then the other (first) Variable can be a number or an array of any shape or size.

Multiplies the two input Variables as $V1 * V2$ on an element-by-element basis. If both input Variables are arrays, then they must both be the same size and shape. If one of the Variables is a number (usually the second input Variable), then the other (first) Variable can be a number or an array of any shape or size.

/

Divides the two input Variables as $V1 / V2$ on an element-by-element basis. If both input Variables are arrays, then they must both be the same size and shape. If one of the Variables is a number (usually the second input Variable), then the other (first) Variable can be a number or an array of any shape or size.

Max

Takes the maximum of the two input Variables on an element-by-element basis. If both input Variables are arrays, then they must both be the same size and shape. If one of the Variables is a number (usually the second input Variable), then the other (first) Variable can be any shape or size.

Min

Takes the minimum of the two input Variables on an element-by-element basis. If both input Variables are arrays, then they must both be the same size and shape. If one of the Variables is a number (usually the second input Variable), then the other (first) Variable can be any shape or size.

Catenate

Both input Variables must be arrays. Merges the two input Variables along their second dimension (columns). The number of rows in both arrays must be equal. For example, if the first input array is of shape (4 x 100) and the second is of shape (4 x 200), then the output array will be of shape (4 x 300) with all columns greater than 100 being those of V2. **Catenate** cannot be used with the **inplace** file output method.

Laminate

Both input Variables must be arrays. Merges the two input Variables along their first dimension (rows). The number of columns of both arrays must be equal. For example, if the first input array is of shape (4 x 100) and the second is of shape (6 x 100), then the output array will be of shape (10 x 100) with all rows greater than 4 being those of V2. **Laminate** cannot be used with the **inplace** file output method.

Convolve

Performs an aperiodic convolution of the two input Variables. If the first input Variable has L columns and the second has M, then the output array will have L+M-1 columns. The order of the arrays is not important. The input Variables may have multiple rows. **Convolve** cannot be used with the **inplace** file output method.

Crosscorr

Computes the crosscorrelation function of the two input Variables. If the first input Variable has L columns and the second has M, then the output array will have L+M-1 columns. The input Variables may have multiple rows. **Crosscorr** cannot be used with the **inplace** file output method.

Quit

Returns to the File Proc Main menu, without performing the processing operation.

8.6 Selection of Input and Output Variables

Before processing starts, you are prompted for the Desk Calculator Variables (R-Z) to use as the first input Variable (*V1*), the second input Variable (*V2*), and output Variable (*V3*). *V2* is only requested if a binary operation was selected.

V2 must already contain the data to be used during processing. If Var > File was selected, *V1* must also already contain the data to be used in processing. All Variables will be overwritten during processing; be sure to save them if you wish to retain their current contents.

Dataset Location	I/O Status														
No dataset specified	<table border="1"> <thead> <tr> <th>Filename</th> <th># Subfiles</th> </tr> </thead> <tbody> <tr> <td>Input: None</td> <td></td> </tr> <tr> <td>Output: C:EKGDMO.DAT</td> <td>13</td> </tr> <tr> <td>Operation: Convolve</td> <td>Binary</td> </tr> <tr> <td>Input var(s): R and S</td> <td></td> </tr> <tr> <td>Output var: T</td> <td>Append</td> </tr> <tr> <td>Current DS: 3</td> <td></td> </tr> </tbody> </table>	Filename	# Subfiles	Input: None		Output: C:EKGDMO.DAT	13	Operation: Convolve	Binary	Input var(s): R and S		Output var: T	Append	Current DS: 3	
Filename	# Subfiles														
Input: None															
Output: C:EKGDMO.DAT	13														
Operation: Convolve	Binary														
Input var(s): R and S															
Output var: T	Append														
Current DS: 3															

I/O variable selection for binary ops	Variables
First input variable (R-Z): (R)	R= 600 REAL
Second input variable (R-Z): (S)	S= .0000
Output variable (R-Z): (T)	T= 600 REAL
Quit/Continue: * C *	U= .0000
(input var 1) {operation} (input var 2)	V= .0000
= (output var)	W= .0000
	X= .0000
	Y= .0000
	Z= .0000

SCREEN R8.5

Input Variable (*V1* - Unary operations)

Select a Desk Calculator Variable (R-Z) to be the first input Variable, *V1*. If a disk file will be used for input, the selected dataset(s) will be automatically and sequentially read into *V1* during processing. Otherwise, the data to be processed must already be stored in *V1*. The processed results are then placed in the output Variable (*V3*).

First input Variable (*V1* - Binary operations)

Select a Desk Calculator Variable (R-Z) to be the first input Variable, *V1*. If a disk file will be used for input, dataset(s) will be automatically read into *V1*

during processing. Otherwise, the data to be processed must already be in *V1*. The data in this Variable will be input as the first operand in the binary operation selected.

Second input Variable (*V2* - Binary operations)

Select a Desk Calculator Variable (R-Z) to be the second input Variable, *V2*. *V2* must already contain the data that is to be input as the second operand to the binary operation selected. Data will not be read into *V2* from a file.

Output Variable (*V3* - Unary & Binary operations)

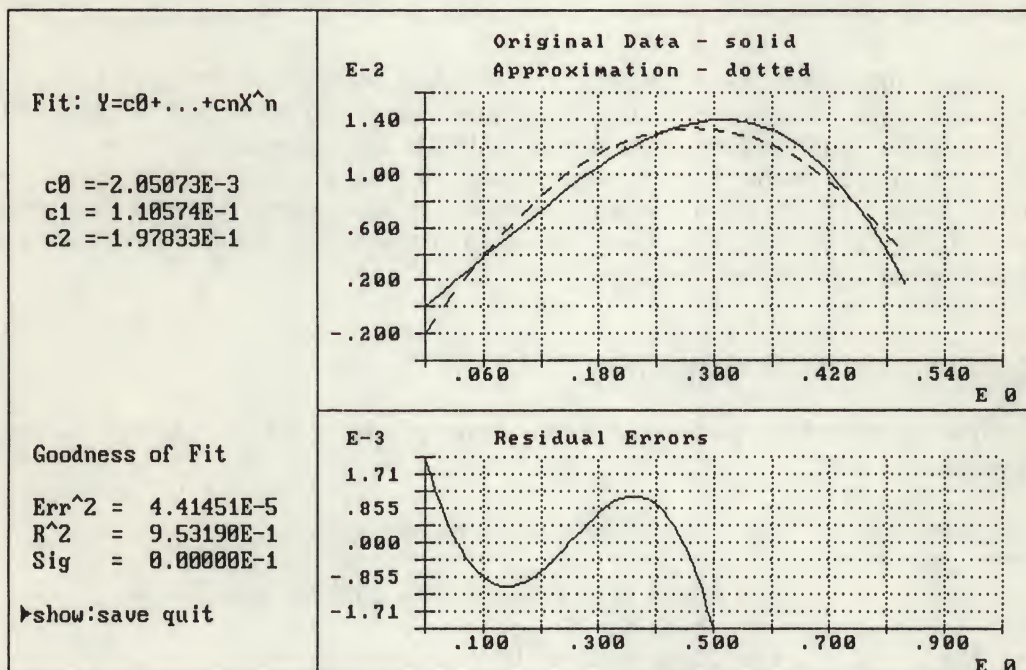
Select a Desk Calculator Variable (R-Z) to be the output Variable, *V3*. This Variable will be used to hold the results of processing *V1* (and *V2*, for binary operations). If a disk file will be used for output, the contents of *V3* will be written to disk as each processed dataset is received. When multiple datasets are output, each output dataset writes over the previous one held in *V3*. Thus, when the operation is completed (or halted), *V3* only contains the last output dataset; previous ones are lost unless they were output to file.

Quit/Continue

Q returns to the File Proc Main menu, without performing the processing operation. **C** begins the processing of data.

Processing may be aborted by pressing **Esc**. There may be a delay between the keystroke and the response because processing is completed for the current dataset before aborting, thereby providing a "clean" interruption.

CHAPTER 9: CURVE FITTING



SCREEN R9.1

9.1 Overview

Curve Fit is used to fit a set of function parameters to give the best least-square fit to an xy-dataset that has been stored in two calculator Variables (R-Z). Fits are available for linear, polynomial, logarithmic, exponential, and multilinear functions; an additional option allows for fitting to an arbitrary user defined function.

Before entering the Curve Fit Main (outer) menu, the user is prompted to specify the two Variables containing the independent (X) values and the dependent (Y) values (or accept the defaults set by the system). For all fit

options *except* **Multilinear** the following must hold:

- * The x and y datasets must both be one dimensional and of the same length.

To fit to a **Multilinear** function:

- * The x dataset must have dimensions $m \times n$ and the y dataset must be one dimensional of length m , where $m > n$ and $n < 10$.

The calculator Variables are scanned for two arrays which comply with either of these requirements. If an appropriate combination cannot be found, a message to that effect is reported on the display and control returns to the Main Menu.

Once the dataset has been chosen, a menu is displayed offering the set of fit options. Each fit option will produce a graph of the original data together with a plot of the fitted curve. The display will also provide a graph of the residuals, a list of the fitted parameters, and several measures of the goodness of fit. Most of the fit options will calculate the fit and produce this display immediately after selection. User Function requires that the user first supply an initial "guess" of the function parameters.

9.1.1 GOODNESS OF FIT PARAMETERS

Each time a fit is performed, three values are reported as measures of the "goodness of fit":

Err ²	--	The sum of the squared residuals
R ²	--	The square of the multiple correlation coefficient.
Sig	--	The significance of fit, computed by applying an F-test to the multiple correlation coefficient.

The formulas used to compute these parameters are different for each fit option and are described in more detail in section 9.2.

9.1.2 ADDITIONAL DIAGNOSTICS AND SAVING RESULTS

Once the solution is displayed, a menu in the lower-left corner of the screen provides options to **show:save** or **quit** (and more in some cases). Selecting **quit** will return processing to the outer Curve Fit menu. Selecting **show:save** will cause a more comprehensive set of diagnostic fit values, including the correlation matrix and standard deviations for the fit parameters, to be listed on the screen [refer to Screen R9.2]. The accompanying prompt list allows the following Curve Fit results to be saved in calculator Variables.

Save Terms		Coefficients	
Save Fitted values in (R-Z): $\langle S \rangle$		Coefficient	Standard Deviation
Save Residuals in (R-Z): $\langle N \rangle$		c1 = -2.05073E-3	3.87565E-4
Save Correlation in (R-Z): $\langle N \rangle$		c2 = 1.10574E-1	3.58476E-3
Save St. Dev.'s in (R-Z): $\langle N \rangle$		c3 = -1.97833E-1	6.93345E-3
Save Poly Coeffs in (R-Z): $\langle N \rangle$			
Press Esc to continue ...			
Goodness of Fit			
Err ² = 4.41451E-5			
R ² = 9.53190E-1			
Sig = 0.00000E-1			
Correlation matrix			
1.0000	.8617	.7417	
.8617	1.0000	.9683	
.7417	.9683	1.0000	

SCREEN R9.2

For each prompt below, the default response, No, will *not* save the corresponding fit values.

Fitted values

Enter the Variable letter (R-Z) in which to store the fitted y-values.

Residuals

Enter the Variable letter (R-Z) in which to store the residuals of the fit. The residuals are the differences between each original y-value and the fitted y-value.

Correlation matrix

Enter the Variable letter (R-Z) in which to store the correlation matrix as it is reported on the display.

Standard Deviations

Enter the Variable letter (R-Z) in which to store the standard deviations as they are reported on the display. The values will be stored as a one- dimensional array.

Poly Coefficients

Only available for **Polynomial** fits. Enter the Variable letter (R-Z) in which to store the calculated polynomial coefficients as they are reported on the display. The values will be stored as a one- dimensional array.

Coefficients

Only available for **Multilinear** fits. Enter the Variable letter (R-Z) in which to store the calculated coefficients as they are reported on the display. The values will be stored as a one- dimensional array.

9.2 Fit Options

Stack Contents	Parameters
(empty stack)	A= .0086 B= 4.3210 C= 2.8693 D= .0000 E= .0000 F= .0000 G= .0000 H= .0000 I= .0000
Curve Fitting	Variables
Array of x-values: (R-Z): (X) Array of y-values: (R-Z): (R) Linear Polynomial Logarithmic Exponential Multilinear User Function Return	R= 51 DP-REAL S= 51 DP-REAL T= 9 DP-REAL U= .0000 V= .0000 W= .0000 X= 51 DP-REAL Y= 51 DP-REAL Z= .0000

SCREEN R9.3

Linear

Computes the a and b parameters of the function $y=ax+b$ that give the best least-square fit to the xy-dataset. (This is equivalent to a first degree polynomial fit -- see below.) The goodness of fit parameters are computed according to the formulas:

$$Err^2 = \sum_{i=1}^n |y_i - (ax_i + b)|^2 \quad \text{EQUATION R9.1}$$

$$R^2 = 1 - \frac{Err^2}{Res^2} \quad \text{where } Res^2 = \sum_{i=1}^n \left| y_i - \frac{1}{n} \sum_{i=1}^n y_i \right|^2 \quad \text{EQUATION R9.2}$$

$$Sig = P_{F[n-2,1]}(f > F) \quad \text{where } F = \frac{R^2(n-2)}{1-R^2} \quad \text{EQUATION R9.3}$$

Polynomial

Computes the coefficients of the polynomial $c_0 + c_1x + \dots + c_nx^n$, that gives the best least-square fit to the xy-dataset. The polynomial can be chosen to be up to 8th degree. The goodness of fit parameters are computed according to the formulas:

$$Err^2 = \sum_{i=1}^n |y_i - (a_mx_i^m + \dots + a_1x_i + a_0)|^2 \quad \text{EQUATION R9.4}$$

$$R^2 = 1 - \frac{Err^2}{Res^2} \quad \text{where } Res^2 = \sum_{i=1}^n \left| y_i - \frac{1}{n} \sum_{i=1}^n y_i \right|^2 \quad \text{EQUATION R9.5}$$

$$Sig = P_{F[n-m-1, m]}(f > F) \quad \text{where } F = \frac{R^2(n-m-1)}{(1-R^2)m} \quad \text{EQUATION R9.6}$$

Logarithmic

Computes the a and b parameters of the function $y = \ln(ax+b)$ that give the best fit to the xy-dataset. The fit is computed by first applying an exponential transformation to the y-values, and then choosing a and b to minimize the weighted least-square error given by the expression

$$Err^2 = \sum_{i=1}^n |e^{-y_i} (e^{y_i} - (ax_i + b))|^2 \quad \text{EQUATION R9.7}$$

Note that the $\exp(-y_i)$ weights are used to compensate for the fact that the exponential transformation will increase the variability (i.e., decrease the accuracy) of the larger data values. (See reference [3] at the end of this chapter.) The goodness of fit parameters apply to the fit of the exponentially transformed data, and are computed as follows:

$$Err^2 = \sum_{i=1}^n |e^{-y_i} (e^{y_i} - (ax_i + b))|^2 \quad \text{EQUATION R9.8}$$

$$R^2 = 1 - \frac{Err^2}{Res^2} \quad \text{where } Res^2 = \sum_{i=1}^n \left| e^{-y_i} \left(e^{y_i} - \frac{\sum_{i=1}^n e^{-y_i}}{\sum_{i=1}^n e^{-2y_i}} \right) \right|^2 \quad \text{EQUATION R9.9}$$

$$Sig = P_{F[n-2,1]}(f > F) \quad \text{where } F = \frac{R^2(n-2)}{1-R^2} \quad \text{EQUATION R9.10}$$

Exponential

Computes the a and b parameters of the function $y = \exp(ax+b)$ that give the best fit to the xy-dataset. The fit is computed by first applying a logarithmic transformation to the y-values, and then choosing a and b to minimize the weighted least-square error given by the expression:

$$Err^2 = \sum_{i=1}^n |y_i (\ln y_i - (ax_i + b))|^2 \quad \text{EQUATION R9.11}$$

If any of the y-values are 0 or negative, the logarithmic transformation cannot be carried out and an error message will be reported. Note that the y_i weights are used to compensate for the fact that the logarithmic transformation will decrease the variability (i.e., increase the accuracy) of the larger data values. (See reference [3] at the end of this chapter.) The goodness of fit parameters apply to the fit of the logarithmically transformed data, and are computed as follows:

$$Err^2 = \sum_{i=1}^n |y_i (\ln y_i - (ax_i + b))|^2 \quad \text{EQUATION R9.12}$$

$$R^2 = 1 - \frac{Err^2}{Res^2} \quad \text{where } Res^2 = \sum_{i=1}^n \left| y_i \left(\ln y_i - \frac{\sum_{i=1}^n y_i^2 \ln y_i}{\sum_{i=1}^n y_i^2} \right) \right|^2 \quad \text{EQUATION R9.13}$$

$$Sig = P_{F[n-2,1]}(f > F) \quad \text{where } F = \frac{R^2(n-2)}{1-R^2} \quad \text{EQUATION R9.14}$$

Multilinear

Computes the coefficients $c_1, \dots, c_n$ that give the best least-square fit to the equation $y = c_1 x_1 + \dots + c_n x_n$, where x_i denotes the i -th column of the x -data set. In order to use this fit option the x -data must be two dimensional, and must have at least as many rows as columns, with a maximum of 9 columns. The y data must be one dimensional with the same length as the x_i 's.

If the x -data is degenerate in the sense that a linear dependency exists between columns, then the calculation will terminate with an error message. Under these circumstances it may be more appropriate to carry out the fit using the **regression** option in the **Stats** menu.

Also note that the multilinear fit does not automatically include a constant term. To force the inclusion of a constant term, simply append a column of 1's to the x -data.

The multilinear goodness of fit parameters are computed using the formulas:

$$Err^2 = \sum_{i=1}^n |y_i - (c_1 x_{i1} + \dots + c_m x_{im})|^2 \quad \text{EQUATION R9.15}$$

$$R^2 = 1 - \frac{Err^2}{Res^2} \quad \text{where } Res^2 = \sum_{i=1}^n |y_i|^2 \quad \text{EQUATION R9.16}$$

$$Sig = P_{F[n-m, m]}(f > F) \quad \text{where } F = \frac{R^2(n-m)}{(1-R^2)m} \quad \text{EQUATION R9.17}$$

User Function

Fits the parameters of a user function to give the best least-square fit to the xy -dataset. When this menu option is executed, the screen is cleared and a window is displayed containing the text

$F(\langle \text{independent variable} \rangle) =$

where the $\langle \text{independent variable} \rangle$ is the Variable selected for the x -values. A flashing cursor is displayed to the right of the $=$, and the user function should be typed in starting at this point. The function should be entered as an algebraic expression, omitting the $\backslash$ that is used in the command mode. The expression should reference the independent Variable, and should use the A-I calculator Parameters for fit parameters. The **Esc** key signals the

completion of function entry, and compiles the expression. If an error is encountered during compilation, it is reported on the display. The user then has the option either to quit or to reedit the equation. Electing to quit will return processing to the outer menu of fit options. The following three examples show legal expressions for user functions.

$$F(X) = \sin(A*X+B)$$

$$F(Z) = \exp(A*Z)+\exp(B*Z)$$

$$F(R) = A/(1+B*(R-C)^2)$$

After the function is compiled, a prompt list is displayed allowing the user to enter estimated values for the (A-I) Parameters. The values entered will replace the current contents of the Parameters. The user must also specify which Parameters are to be fitted (at least one Parameter must be selected). This prompt list also allows for modification of the following three aspects of the fitting procedure:

Fit Type

Algorithm used for the fitting process. Options are Gauss-Newton, BFGS, or a Hybrid of the two.

Maximum Iterations

The maximum number of iterations allowed before terminating the fit.

Fit Tolerance

Accuracy factor used to determine when the parameters are accurate enough to terminate the fit.

These parameters control the way that the fit operation is carried out, and are described in more detail in section 9.3.

Once these parameters have been selected, a graphics display is produced that is identical to the ones generated by the other fit options. The initial "guesses" for the fit parameters are used for the fitted function. In the lower-left corner of the display a menu is shown offering the following options:

fit

Apply the fitting procedure to improve upon the current values of the fit parameters. The parameter values are adjusted, and the display is updated to reflect the effects on the fit.

pedit

Edit the current values of the calculator Parameters and fit specifications. **Pedit** is typically used to make adjustments to the initial guesses before invoking **Fit**.

show:save

Display a more extensive set of fit diagnostics, including the asymptotic correlation matrix and standard deviations for the fit parameters. This option also allows for saving curve fit results in the calculator Variables.

quit

Return to the outer curve fit menu.

The goodness of fit parameters are computed using the formulas:

$$Err^2 = \sum_{i=1}^n |y_i - F[x_i]|^2 \quad \text{EQUATION R9.18}$$

$$R^2 = 1 - \frac{Err^2}{Res^2} \quad \text{where } Res^2 = \sum_{i=1}^n |y_i|^2 \quad \text{EQUATION R9.19}$$

$$Sig = P_{F[n-m,m]}(f > F) \quad \text{where } F = \frac{R^2(n-m)}{(1-R^2)m} \quad \text{EQUATION R9.20}$$

9.3 User Function Fits

9.3.1 CURVE FITTING ALGORITHMS

This section discusses the three algorithms available for fitting user functions.

The problem is stated using a vector notation: let $\mathbf{c}=(c_1,\dots,c_m)$ be the vector of fit coefficients, $\mathbf{x}=(x_1,\dots,x_n)$ be the vector of values for the independent variable, and $\mathbf{y}=(y_1,\dots,y_n)$ be the vector of dependent values. We also use the notation that $F(\mathbf{c},\mathbf{x})$ is the vector of values produced by the user function when it is applied in succession to the x_i 's. The square error can then be written in the form:

$$E(\mathbf{c}) = \sum_{i=1}^n |y_i - F(\mathbf{c};x_i)|^2 = \|\mathbf{y} - F(\mathbf{c},\mathbf{x})\|^2 \quad \text{EQUATION R9.21}$$

Next suppose that $\mathbf{c_inf}$ is the vector of optimal values for the c_i 's. By modeling $E(\mathbf{c})$ as a quadratic centered at $\mathbf{c_inf}$ (i.e., ignoring higher order terms) we can write :

$$E(\mathbf{c}) = C + (\mathbf{c} - \mathbf{c}_\infty)A(\mathbf{c} - \mathbf{c}_\infty), \quad \text{EQUATION R9.22}$$

where C is a constant and A is an m by m symmetrical matrix. A simple computation then gives us:

$$\begin{aligned} \text{Grad } E(\mathbf{c}) &= 2A(\mathbf{c} - \mathbf{c}_\infty) \text{ and} \\ \text{Hess } E(\mathbf{c}) &= 2A \end{aligned} \quad \text{EQUATION R9.23}$$

These equations can then be combined to produce the formula :

$$\mathbf{c}_\infty = \mathbf{c} - [\text{Hess } E(\mathbf{c})]^{-1} \text{Grad } E(\mathbf{c}). \quad \text{EQUATION R9.24}$$

According to this formula, the vector $\mathbf{c}$ differs from the vector of optimal parameters by the term :

$$[\text{Hess } E(\mathbf{c})]^{-1} \text{Grad } E(\mathbf{c}).$$

EQUATION R9.25

Of course, this relationship is only exact for the quadratic model, but the above term can still be used as a correction factor in the general case. This is known as the Newton-Raphson method for minimization. In this case E is the function being minimized. This is the basic approach that is used in all three fitting algorithms. Actually the direction of the correction vector is of more importance than its exact magnitude, since a line search is performed to minimize error along that direction.

The difficulty in directly applying the Newton-Raphson method is that it is extremely time consuming to compute the Hessian matrix of E ; and although it takes fewer iterations of this method to arrive at the optimal parameters, it is generally more efficient to take coarser steps that can be computed more rapidly.

Most standard approaches to non-linear curve fitting use the Newton correction factor with an approximate rather than exact evaluation of the Hessian matrix. The main difference between the methods is in the manner of approximation.

In the Gauss-Newton method the approximation is obtained by taking the Hessian of the right hand side of Eq. R9.21 and then discarding some of the factors. Expanding the Hessian gives

$$\begin{aligned} \text{Hess } E(\mathbf{c}) &= \text{Hess } [(y - F(\mathbf{c}, x)) \cdot (y - F(\mathbf{c}, x))] \\ &= 2[\text{Jacob } F(\mathbf{c}, x)]' [\text{Jacob } F(\mathbf{c}, x)] \\ &\quad - 2[\text{Hess } F(\mathbf{c}, x)] (y - F(\mathbf{c}, x)), \end{aligned}$$

EQUATION R9.26

and the second term is discarded. A heuristic argument for this is that the second term is proportional to the size of the error vector $(y - F(\mathbf{c}, x))$, which grows smaller and smaller as the solution is approached. Unfortunately, this is not the case if the data is noisy. Furthermore, the remaining term is sometimes nearly rank deficient which can result in serious numerical inaccuracies. Nevertheless, the Gauss-Newton method is probably the most efficient for the majority of applications.

The second algorithm is called the BFGS method. It approximates the Hessian by making use of past values of $\mathbf{c}$ and $\text{Jacob } F(\mathbf{c})$. When the curve fitting procedure begins, the Hessian is guessed to be the identity matrix. Then after each iteration, this guess is improved by using the difference between the

Jacobian before and after c is modified (see reference [2], in this source the BFGS-method is called the complementary DFP formula). The BFGS method is generally slower than the Gauss-Newton method, but it will converge to the solution in cases where the Gauss-Newton method does not. It is also much less susceptible to the problem of rank deficiency.

The hybrid method is a combination of the Gauss-Newton and BFGS methods. A Levenberg-type modification (see p. 278 of reference [1]) is applied to the Gauss-Newton approximation, where the modifying component is a multiple of the BFGS-approximation rather than the identity matrix used in the Levenberg-Marquandt approach. The size of the modifying component is adapted as the fit progresses. Thus, if the Gauss-Newton approximation becomes more accurate, then the modifying term will recede.

The best approach to using these methods is probably to try the Gauss-Newton method first. If that does not work, then turn to one of the other two.

9.3.2 TROUBLESHOOTING USER FITS

General non-linear curve fitting is a very complicated process. It can easily fall prey to problems of numerical inaccuracies, and can also fail to converge if the original problem is incorrectly specified. In this section several suggestions are offered to help make the best use of the User function option.

* *The fit parameters should be independent.* In mathematical terms this means that the columns of the Jacobian matrix should be linearly independent. On a more elementary level, this means that a change in each fit parameter should have an effect on the user function that cannot be emulated by changing the other variables. Some examples of user functions that have parameters that are not independent are:

$$y=(a+b)*x+c$$

$$y=a*\exp(b*x+c).$$

In the first equation the a and b parameters both have the same effect. This is also true of the a and c parameters in the second equation, although here it is less evident. One test that can be used to determine if parameters are interdependent is to see if it is possible to express the same function using two different sets of parameter values. For instance, if the parameters a , b , and c are assigned the values $a=2$, $b=3$, and $c=0$, then the second equation will simplify to the form $y=2\exp(3x)$. But this same result will also be produced by the assignments $a=1$, $b=3$, and $c=\ln(2)$.

In complicated expressions it is sometimes difficult to determine if all the fit parameters are independent. In such cases it is helpful, and also computationally more efficient, to simplify as much as possible.

* *The fit parameters should be scaled to have the same orders of magnitude.* In the course of the fitting process, many calculations involve a mixture of the

fit parameters. To avoid numerical errors it is a good practice to set up the fitting function so that the fit parameters will have roughly the same orders of magnitude. For instance, exponential decay expressions of the form

$$a \cdot \exp(-b \cdot x)$$

will generally use very large values for the a parameter (say on the order of 10,000), and very small values for the decay rate b (e.g., 0.001). The ranges of the two parameters can be normalized by replacing the above term with:

$$a \cdot 10000 \cdot \exp(-b \cdot 0.001 \cdot x).$$

* *The fit parameters should be double precision.* Part of the fit computation requires numerical differentiation of the user function with respect to each fit parameter. This means that differences in the function values must be measured for slightly perturbed values of the fit parameters. To assure that this calculation is accurate it is best to set up the fit parameters to be double precision.

* *Use a restricted data set if the computation takes too long, or if it exceeds the available memory.* If the x and y data sets contain several hundred elements, it may be more expeditious to use every other entry, or even every third entry for calculating the fit. Alternatively, the xy -data set can also be reduced in size by averaging over adjacent pairs of data points.

References

- [1] D. A. Jacobs, ed. *The State of the Art in Numerical Analysis*. New York: Academic Press, 1977.
- [2] G. R. Walsh. *Methods of Optimization*. New York: John Wiley, 1975.
- [3] Philip R. Bevington. *Data Reduction and Error Analysis for the Physical Sciences*. New York: McGraw-Hill, 1969.

CHAPTER 10: **POLYNOMIAL OPERATIONS**

10.1 Overview

10.1.1 GENERAL DESCRIPTION

The **Polys** option is used to set up and manipulate the 10 polynomials (P1-P10) that are available as functions in the desk calculator. Each polynomial can be up to ninth degree, and can have either real or complex coefficients. Polynomial arithmetic can be carried out on each polynomial, Pn, including the operation of synthetic division. The Pn's can also be integrated, differentiated, and shifted, and a **roots** option is provided that will find the complete set of roots of an arbitrary real or complex polynomial. The **Polys** menu also has options for automatically setting up Legendre, Laguerre, Hermite, and Tchebyshev polynomials.

10.1.2 THE POLYNOMIAL DISPLAY

In the upper right hand portion of the display, the Pn's are listed along with their current dimensions. One entry in this list is displayed in inverse, and the coefficients of that entry are shown in the upper left. The **PgUp** and **PgDn** will move the inverted entry up and down the list, thereby allowing the coefficients of any polynomial to be recalled to the screen. The coefficients may be changed via the **Edit** option.

10.1.3 USING POLYNOMIALS

Once a polynomial is defined through the **Polys** option, it may be invoked by typing **Pn** on the command line, where n is its number, 1-10. The polynomial function will be applied to the top stack entry and will reference the current Parameter values if any Parameters were specified in the equation. A polynomial function may also be invoked within a **User Function** by typing **Pn** in its command line definition (see Chapter 3).

10.2 The Menu of Polynomial Operations

The main menu of polynomial operations is displayed in the lower left corner of the screen. It's menu options use the "work area" in the lower right hand corner for prompt lists. After a menu option is executed, the polynomial list on the top right of the screen is repainted to reflect any changes, and the polynomial whose coefficients are shown on the top left is switched to the P_n that was modified by the menu option.

Add

Adds together two polynomials, and saves the result in a third.

Sub

Subtracts one polynomial from a second, and saves the result in a third.

Mult

Multiplies two polynomials and saves the result in a third.

Div

Divides one polynomial by a second, saving the quotient in a third, and the remainder in a forth.

Shift

Shifts a polynomial by a specified factor, and saves the result in another polynomial. If $P(x)$ is the input polynomial, and a is the shift factor then the output polynomial will be $P(x+a)$.

Edit

Allows editing of polynomial coefficients. Prompts also allow switching between real and complex coefficients.

Coeffs>Var

Allows copying of the coefficients of a polynomial into one of the (R-Z) system Variables.

Roots

Extracts the complete set of roots of a selected polynomial. The extraction process is iterative, so a tolerance factor must be entered to control the accuracy of the computation. Note that making the tolerance factor too small sometimes introduces serious numerical errors. The roots can also be saved in a selected system Variable (R-Z).

Integrate

Integrates one polynomial and saves the result in another. Integration cannot be performed on polynomials of ninth order because the integral would be tenth order, which exceeds the limits of the Pn's.

Differentiate

Differentiates one polynomial and saves the results in another.

Legendre

Generates a Legendre polynomial and assigns it to one of the Pn's. The Legendre polynomials are defined by the formula

$$P_n(z) = \frac{1}{2^n n!} \frac{d^n}{dz^n} (z^2 - 1)^n$$

Laguerre

Generates a Laguerre polynomial and assigns it to one of the Pn's. The Laguerre polynomials are defined by the formula

$$L_n(z) = \sum_{m=0}^n (-1)^m \binom{n}{m} \frac{1}{m!} z^m$$

Tchebyshev

Generates a Tchebyshev polynomial and assigns it to one of the Pn's. The Tchebyshev polynomials are defined by the formula

$$T_n(z) = \frac{n}{2} \sum_{m=0}^{\left[\frac{n}{2} \right]} (-1)^m \frac{(n-m-1)!}{m! (n-2m)!} (2z)^{n-2m}$$

with $[k]$ = smallest integer part of k

Hermite

Generates a Hermite polynomial and assigns it to one of the Pn's. The Hermite polynomials are defined by the formula

$$H_n(z) = n! \sum_{m=0}^{\left[\frac{n}{2} \right]} (-1)^m \frac{1}{m! (n-2m)!} (2z)^{n-2m}$$

with $[k]$ = smallest integer part of k

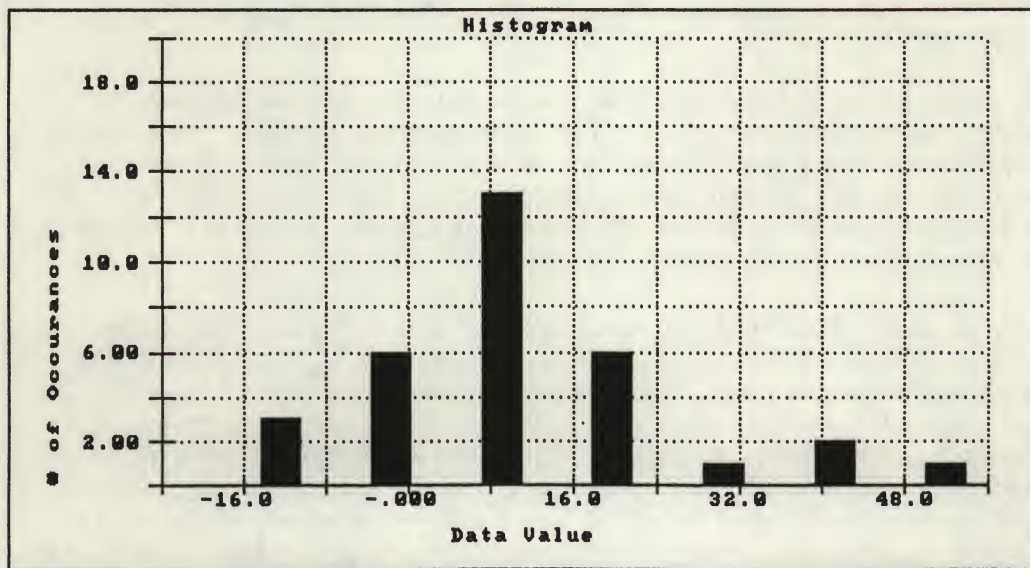
Var>Coeffs

Allows reading of polynomial coefficients from one of the (R-Z) system Variables.

Return

Returns to the Desk Calculator Main Menu.

CHAPTER 11: STATISTICS



Histogram
Z= 4 x 8 REAL
Start Column: 1 End Column: 8
Start Row: 1 End Row: 4
To Plotter
Save on Disk
Return

SCREEN R11.1

11.1 Overview

The Stats option contains procedures that perform statistical analyses on tabular data. These procedures can be categorized as follows:

- * descriptive statistics
- * sorting
- * theoretical distributions
- * hypothesis testing (parametric and nonparametric)
- * multivariate stepwise regression
- * analysis of variance (ANOVA)
- * principal-component analysis

Data to be used in these analyses can be taken from calculator Variables, or may be entered manually with a built-in array editor.

The *descriptive statistics* available in Stats are **Basic Stats**, **Percentiles**, and **Histogram**. These operations obtain information about the distribution of data values. In each case, the data can be restricted to a rectangular subsection of an input Variable. The basic statistics computed by **Basic Stats** are listed on the display, but can also be output to the printer or note file. Information about the cumulative distribution function of the current data values is obtained with **Percentiles**. The **Histogram** routine allows you to construct a histogram which can be saved in a Variable and plotted on the screen or plotter. A plot may also be saved in a disk file for later recall in the **Graphics** menu (see recall in Section 4.2).

A row or column of a block of tabular data may be *sorted* with the **Sort** option. The other rows (or columns) will be permuted in the same order as the sorted row (or column).

Percentages and percentiles of four of the most common cumulative *theoretical distributions* can be evaluated by selecting **Distrib's**. This feature replaces the tables found in the back of most statistics books. These distributions are: normal, Chi-square, Student t, and F(n,m). Where appropriate, the mean and standard deviation may be entered to eliminate the need to convert to normalized data values.

The menu item **Hyp tests** opens a list of parametric and nonparametric *hypothesis tests* for location and dispersion (spread). The parametric tests include one-sample t tests, two-sample, paired and unpaired t tests, the Chi-square test for one-sample variance, and the F test for two-sample equality of variance. The nonparametric tests include the Kolmogorov-Smirnov test for normality, the Wilcoxon signed-rank test, and the Mann-Whitney rank-sum test. The data to be subjected to testing can be chosen from any of the Desk Calculator Variables. Furthermore, alternative hypotheses appropriate for the test can be chosen from a set supplied with each test.

An interactive *stepwise regression* routine is available with the selection of **Regression**. The independent (X) data array and the dependent (Y) data array can be selected from the Desk Calculator Variables. The fit variables (columns 1, 2, etc. of (X)) may be entered or removed from the model one at a time, until the desired model is found. The fit results can be saved in Variables at the end of the procedure.

Three different types of *ANOVAs* can be carried out: **1 way ANOVA**, used for analyzing the variations within one column, according to the categories of another column, **2 way ANOVA**, used for analyzing variations in one column according to the categories of two columns, and **table ANOVA**, used for analyzing the variations of an entire table according to classification by row and column.

The correlation and/or covariance matrix of a set of Variables can be found with the **Correlation** menu item. A *principal-component analysis* can then be performed on either of these two matrices and the resulting components (eigenvectors) and eigenvalues can be saved in Desk Calculator Variables.

11.2 The Statistics Main Menu

Basic Statistics				
Select variable (R-Z): (X)			Maximum:	8.3000
Start	End		Minimum:	.1200
row: (3)	(5)		Sum:	17.8780
col: (2)	(3)		Mean:	2.9797
			Median:	1.4640
			Sample Variance:	12.1494
			Standard Deviation:	3.4856
			Skewness:	.9290
			Kurtosis:	-1.1620
			Sum of Squares:	114.0173
			Root Mean Square:	4.3592
Continue/Quit/New Parms: * Q *				
Menu Options		Variables		
Edit	Basic Stats	R= 3 DP-REAL		
Sort	Percentiles	S= 8 x 2 REAL		
Hyp Tests	Histogram	T= 2 x 2 REAL		
Distrib's	ANOVA	U= .0000		
Correlation	Regression	V= .0000		
Return		W= .0000		
		X= 8 x 5 REAL		
		Y= 8 x 1 REAL		
		Z= 4 x 8 REAL		

SCREEN R11.2

Edit

Allows modification or creation of the entries in a data table. For modification the table is chosen from an existing calculator Variable, whereas for creation an empty Variable is chosen. The table is displayed with a highlighted element that can be scrolled through the array using the arrow keys. The value of this element can be changed at any time by typing a number followed by **Enter**. Each time a number is entered, the highlighted element is advanced one step in the same direction as the most recently pressed arrow key. Columns and rows may be copied and deleted, thus providing adjustable, interactive sizing of the table. The edit mode is terminated by pressing the **Enter** without having typed in a number. The Edit option is identical to the **aedit** option in the Desk Calculator (see Chapter 2.)

Basic Stats

Computes the following set of basic statistics for any rectangular subset of the selected Variable:

Maximum	Standard Deviation
Minimum	Skewness
Sum	Kurtosis
Mean	Sum of Squares
Median	Root Mean Square
Variance	

Options are provided to output the results to either the printer or the note file.

Sort

Allows a row or column of any selected Variable to be sorted with the remaining rows or columns being permuted exactly as the sorted row or column. When a row is selected, the column prompt automatically defaults to -1 to indicate that the sort will be performed on the row. The converse occurs when a column number is selected.

Percentiles

Computes percentiles for any rectangular subset of the selected Variable. The user is prompted for the percentage and is given the corresponding percentile.

Hyp Tests

Opens a menu of various hypothesis tests to be performed on a selected Variable(s). The menu includes some of the most common parametric and nonparametric tests. See section 11.3.

Histogram

Generates a histogram for any rectangular subset of the selected Variable. The number of breakpoints in the histogram must be chosen to be between 2 and 100, and the histogram "bins" are then set up to be equally spaced between the minimum and maximum data values. Thus, for example, if the data ranges between 0 and 1, and if two breakpoints are selected, then there will be three bins with the ranges: $[0, 1/3]$, $(1/3, 2/3]$, and $(2/3, 1]$. The computed histogram and its breakpoint array may be saved in calculator Variables for further analysis. Three options are presented before the histogram is generated:

Generate

Generates the histogram and stores the results in the calculator Variables specified.

Plot

Generates the histogram and plots the result on the display. A submenu provides these additional options:

- | | |
|----------------|--|
| Plotter | -- Reproduces the display on the plotter. |
| Save | -- Copies the display to a disk file that can be recalled to the full-screen window in Graphics . |
| Return | -- Saves the results in calculator Variables, if specified, and returns to the Histogram menu. |

Quit

Returns to the Statistics Main menu.

Distrib's

Opens another menu for the computation of theoretical percentages and percentiles of the normal, Chi-square, Student t, and F(n,m) distributions. See section 11.4.

ANOVA

Opens another menu enabling one-way, two-way, and table analyses of variance to be performed on data. See section 11.5.

Correlation

Computes the correlation and/or covariance matrices of the selected table and saves the results in calculator Variables, if specified. The saved correlation or covariance matrix may then be subjected to principal component analysis (i.e., eigenvalue-eigenvector decomposition) and these results may also be saved in Variables.

Regression

Performs a stepwise regression analysis. The dependent data values {y} are modeled as a linear sum of several independent parameters {x1, x2, ... , xN}, with fit coefficients determined by a least-square analysis. See section 11.6.

Return

Returns to the Desk Calculator Main Menu.

11.3 Hypothesis Testing

Kolmogorov-Smirnov Test		Test specifications
Select variable (R-Z): (Z)		Significance level: (.0100)
Start	End	Less/Not equal/Greater: (N)
row: (1)	(4)	Quit/Continue: * Q *
col: (1)	(8)	
		H0: Distribution is normal
		H1: Distribution is not normal
Continue/Quit/New Parms: * C *		
Kolmogorov-Smirnov Results		
Var 1 stats		
Mean: 10.2937		
St dev: 15.7554		
# points: 32		
Test stat D: .1316		Accept H0: Distribution is normal
P(Z > D): > 0.2		at significance level: .0100

SCREEN R11.3

The Hyp Tests menu allows hypothesis testing to be performed on selected tables of data. The menu includes both parametric and nonparametric tests. The parametric tests should be used if the data is normally distributed, since these tests are more powerful than their nonparametric counterparts. Thus, the first hypothesis test should generally be the Kolmogorov-Smirnov test of normality. Note that this test is nonparametric and requires a continuous distribution for the data being tested for normality.

If the data is not normal, then the nonparametric tests can be used, provided that the sample distribution is continuous and symmetric about its median.

References [1] and [2], found at the end of this chapter, explain in greater detail the nature and use of the tests performed in this menu.

KS Normality test

Performs the Kolmogorov-Smirnov normality test on any rectangular subset of the selected Variable. The significance level can be set to any value between 0.0 and 0.2. The alternative hypothesis can be chosen to be skewed right (Less than), Not equal, or skewed left (Greater than). The value of the test statistic, $D (= \max[D-, D+])$, is printed along with its associated significance level and a message relaying the acceptance or rejection of the null hypothesis.

1 Sample t test

Tests the mean of any rectangular subset of a Variable. A hypothesized mean, significance level, and alternative hypothesis can be chosen. The significance level must be less than 0.2. The alternative hypothesis can be that the true mean is Less than, Not equal, or Greater than the hypothesized mean. The test statistic, t , is printed with its associated significance level, degrees of freedom, and a message relaying the acceptance or rejection of the null hypothesis. Note that the test assumes normally distributed data.

2 Sample t test

Tests the equality of means from two sets of data. Data can be chosen from any rectangular subset of selected Variables. The test can be either a paired or unpaired test. The significance level must be set less than 0.2. The alternative hypothesis can be either that the mean of Variable 1 is Less than, Not equal, or Greater than that of Variable 2. The test statistic, t , is printed with its associated significance level, degrees of freedom, and a message relaying the acceptance or rejection of the null hypothesis. Note that the test assumes equal variances for the two sets of data. Also note that the test assumes normally distributed data.

1 Sample Chi-sq test

Tests the value of the data variance against an hypothesized value. Any rectangular subset of the selected Variable can be tested. The hypothesized variance, significance level, and alternative hypothesis can be chosen. The significance level must be less than 0.2. The alternative hypothesis can be that the true variance is Less than, Not equal, or Greater than the hypothesized variance. The test statistic, X^2 , is printed with its associated significance level and a message relaying the acceptance or rejection of the null hypothesis. Note that if the alternative hypothesis is that the true and hypothesized variances are not equal, the label " $P(|Z| > X^2)$ " gives the significance level for the test statistic being less than or greater than the critical values associated with this significance level. Also note the test assumes normally distributed data.

2 Sample F test

Tests the equality of variances from two sets of data. The data can be chosen from any rectangular subset of selected Variables. The significance level must be less than 0.2. The alternative hypothesis can be either that the variance of Variable 1 is Less than, Not equal, or Greater than that of Variable 2. The test statistic, F , is printed with its associated significance level, degrees of freedom, and a message relaying the acceptance or

rejection of the null hypothesis. Note that if the alternative hypothesis is that the two variances are not equal, the label " $P(|Z| > F)$ " gives the significance level for the test statistic being less than or greater than the critical values associated with this significance level. Note the test assumes normally distributed data.

Wilcoxon Signed-rank test

Tests the median of any rectangular subset of a Variable against an hypothesized value. The hypothesized median, significance level, and alternative hypothesis can be chosen. The significance level must be less than 0.2. The alternative hypothesis can be that the true median is Less than, Not equal, or Greater than the hypothesized median. The test statistic, T_+ , is printed with its associated significance level and a message relaying the acceptance or rejection of the null hypothesis.

Note that this test assumes that the data distribution is symmetric and continuous and that no two data values are equal.

Mann-Whitney Rank-sum test

Tests the equality of medians from two sets of data. The data can be chosen from any rectangular subset of selected Variables. The significance level must be less than 0.2. The alternative hypothesis can be either that the median of Variable 1 is Less than, Not equal, or Greater than that of Variable 2. The test statistic, U , is printed with its associated significance level and a message relaying the acceptance or rejection of the null hypothesis.

Note that this test assumes that the data distribution is symmetric and continuous and that no two data values are equal.

Return

Returns to the Stats Main menu.

11.4 Theoretical Distributions

Theoretical Distributions		Percentages of the F[n,m] Dist.	
Normal %'S Normal %tiles Chi Sq %'S Chi Sq %tiles Student t %'S Student t %tiles F %'S F %tiles Return		Deg of Freedom n: (3) Deg of Freedom m: (2) Value: (.9000) Percentile: 43.5410 Press Esc to continue	
Menu Options		Variables	
Edit Sort Hyp Tests Distrib's Correlation Return		Basic Stats Percentiles Histogram ANOVA Regression	
		R= 51 DP-REAL S= 51 DP-REAL T= 9 DP-REAL U= .0000 V= .0000 W= .0000 X= 51 DP-REAL Y= 51 DP-REAL Z= .0000	

SCREEN R11.4

The **Distrib's** menu computes the percentages and percentiles of four of the more common distributions. The menu items in this menu replace the probability tables frequently found in the back of statistics reference books.

Normal %'s

Computes percentages of the Normal distribution. The user selects the standard deviation, the mean, and the value and is returned the percentile for that value.

Normal %tiles

Computes percentiles of the Normal distribution. The user selects the standard deviation, the mean, and the percentile and is returned the value for that percentile.

Chi Sq %'s

Computes percentages of the Chi Squared distribution. The user selects the degree of freedom and the value and is returned the percentile for that value.

Chi Sq %tiles

Computes percentiles of the Chi Squared distribution. The user selects the degree of freedom and the percentile and is returned the value for that percentile.

Student t %'s

Computes percentages of the Student t distribution. The user selects the degree of freedom and the value and is returned the percentile for that value.

Student t %tiles

Computes percentiles of the Student t distribution. The user selects the degree of freedom and the percentile and is returned the value for that percentile.

F %'s

Computes percentages of the $F(n,m)$ distribution. The user selects the degree of freedom n , the degree of freedom m , and the value and is returned the percentile for that value.

F %tiles

Computes percentiles of the $F(n,m)$ distribution. The user selects the degree of freedom n , the degree of freedom m , and the percentile and is returned the value for that percentile.

11.5 Analysis of Variance

ANOVA menu		Table Analysis of Variance		
1 Way ANOVA 2 Way ANOVA Table ANOVA Return		Table (R-Z): (Z) Continue/Quit/New Parms: * Q *		
Sum Sq's	Deg Freedom	Mean Sum Sq's	F-Value	Significance
5516.5640	3	1838.8547	29.7314	.0000
879.8088	7	125.6870	2.0322	.0987
1298.8262	21	61.8489	.0000	.0000

SCREEN R11.5

An analysis of variance (ANOVA) separates the variations of a dependent variable into groups defined by a set of discrete measurement categories. The analysis seeks to determine if the overall variations in the dependent variable can be partially explained by these categories. The determination is done by comparing the sum of squares for the categories with the residual sum of squares.

Three types of ANOVAs are available in this menu: one-way, two-way, and table. The one-way ANOVA deals with experiments that classify the dependent variable over a single experimental factor, e.g., mile/gallon measurements as a function of carburetor type. The two-way ANOVA deals with experiments that classify the dependent variable over two experimental factors, e.g., plant growth rate as a function of fertilizer type and artificial lamp type. Note that in the two-way ANOVA model, a portion of the dependent variable variations will be due to the combined effects of factor 1 and factor 2, i.e., a (factor 1/factor 2) component is considered. The table ANOVA differs from the two-way ANOVA in that it should be used when there is only one data value per dual classification (cell), e.g., only one measurement was made for each combination of fertilizer type and lamp type. Also, the model for the table ANOVA does not include a (factor 1/factor 2) interaction term.

1 way ANOVA

Generates a one way analysis of variance chart for the variations in one column vector of data, classified by the entries of another vector. Vectors may either be one dimensional Variables or columns selected from two dimensional Variables. Elements in the sample vector will be grouped together if their corresponding entries in the factor vector are equal. On completion, a table is printed showing statistics of the model in the top row and the residual statistics in the bottom row. From left to right, the columns contain:

- | | |
|-------|--|
| Col 1 | The sum of squares for portion of the data explained by the model and the residual sum of squares. |
| Col 2 | The degrees of freedom associated with the model and the residuals. |
| Col3 | The mean sum of squares (sum of squares divided by degrees of freedom). |
| Col 4 | The F value for the model, i.e., the ratio of mean sums of squares. |
| Col5 | The significance level for the F value, i.e., the probability that F could be greater than it is. |

2 way ANOVA

Generates a two way analysis of variance chart for the variations in a column vector of the data, classified by the entries of two other columns. All three vectors may be selected from either one dimensional Variables or columns of two dimensional Variables. Elements in the sample vector will be grouped together if their corresponding entries in the factor vectors are equal. On completion, a table is printed showing statistics of the factor 1 component in the top row, factor 2 component in the next row, the combined (factor 1/factor 2) component in the third row, and the residuals in the bottom row. From left to right, the columns contain:

- | | |
|-------|---|
| Col 1 | The sum of squares for the portion of the data explained by factor 1 of the model, the factor 2 sum of squares, the combined (factor 1/factor 2) sum of squares, and the residual sum of squares. |
| Col 2 | The degrees of freedom associated with the model factors and the residuals. |
| Col 3 | The mean sum of squares (sum of squares divided by degrees of freedom). |

- Col 4** The F value for factor 1 of the model, i.e., the ratio of the mean sum of squares for factor 1 and the residual sum of squares, the F value for factor 2 of the model, and the F value for the combined (factor 1/factor 2) portion of the model.
- Col5** The significance level for the F value of factor 1, i.e., the probability that the factor 1 F could be greater than it is, the significance level for the F value of factor 2, and the significance level for the combined (factor 1/factor 2).

Table ANOVA

Generates an analysis of variance chart for the variations in the data array, classified by the row (factor 1) and column (factor 2) indices. This menu option can only be used when the data array is two dimensional. On completion, a table is printed showing statistics of the factor 1 component in the top row, the factor 2 component in the next row, and the residuals in the bottom row. This table is essentially the same as that for the two-way ANOVA, except that no (factor 1/factor 2) information is given since there are not enough data values to determine the effects of such a combination.

11.6 Stepwise Regression

In Model Vars	Value	St Dev	F-to-rem	Sig	Overall Model
2	.2841	.1330	4.5609	.0858	Deg's freedom
3	1.2778	1.0952	1.3612	.2959	2
					Residual SS
					3.5922
					Regression SS
					2.792E2
Not In Vars	Part corr	Tol	F-to-ent	Sig	y intercept
1	-.0654	.0691	.0172	.9020	.2122
4	-.6173	.0031	2.4626	.1917	r^2 corr coeff
5	-.3195	.7595	.4547	.5371	.9873
					F-all value
					1.943E2
					Significance
					1.819E-5

Enter/Remove: (E)
Variable x (4)

Quit/Continue: * C *

SCREEN R11.6

Stepwise regression provides a means to determine the most suitable model to describe the variations observed in a dependent variable y as a linear function of a set of independent variables $x_1, x_2, \dots, x_N$. This model is of the form

$$y = c_0 + c_1 * x_1 + c_2 * x_2 + \dots + c_N * x_N$$

EQUATION R11.1

In general, variations in $\{y\}$ will be more closely correlated with variations in some of the elements of $\{x_i\}$ than with others. The purpose of stepwise regression is to determine this subset of $\{x_i\}$ in a statistically meaningful manner. Discussions on the theory behind stepwise regression can be found in References [3] and [4]; for convenience, a brief overview will be given next.

NOTE

Session 6 of the Tutorial guides you through a sample regression analysis. For additional information on how to use the **Statistics Regression** option, we recommend that you follow the step-by-step instructions found there.

11.6.1 OVERVIEW OF THEORY

The regression analysis is actually done on the reduced {y} and {xi} values obtained by subtracting the mean values from each variable; thus the reduced model is:

$$y' = c_1 * x_1' + c_2 * x_2' + \dots + c_N * x_N' \quad \text{EQUATION R11.2}$$

where the prime marks indicate that the appropriate mean value has been subtracted. Note that the bias coefficient c_0 is absent from this model, but it can be computed by subtracting Eq. R11.2 from Eq. R11.1.

It is important to note that the stepwise computations are done under the assumption that the variances of each of the data points in the set of {y} measurements are equal: less than optimal results will be obtained if the variances are not equal.

The stepwise regression display includes a table of values relating the significance of entering each x_i . This table of variables not in the model contains the following list of information:

1. the square of the partial correlation of each x_i with respect to the dependent variable, denoted as $r^2\text{-to-ent}$ (r^2 to enter);
2. the tolerance of each x_i (one minus the squared multiple correlation of each x_i with respect to the previously entered x_i 's) denoted as Tol;
3. the ratio of the net decrease of the residual sum of squares for each x_i over the overall residual sum of squares if that x_i was in the model, weighted by the appropriate degrees of freedom, denoted as $F\text{-to-ent}$ (F-to-enter);
4. the significance level for the current F-to-enter for each x_i .

This list aids in the selection of the one independent variable not in the model that most reduces the residual sum of squares. Once several independent variables are in the model, it may be appropriate to see the effect of removing a variable from the model. The table of variables in the model includes the following information:

1. the value of the fit coefficient for each x_i ;
2. the standard deviation of the fit coefficient for each x_i ;
3. the ratio of the net increase of residual sum of squares for each x_i over the overall residual sum of squares if that x_i were not in the

model, weighted by the appropriate degrees of freedom, denoted as **F-to-rem** (F-to-remove);

4. the significance level for the F-to-remove for each x_i .

This list aids in the choice of that variable which when removed would cause the smallest increase in the residual sum of squares.

Information of the overall model fit is also displayed. This information includes the values of the residual sum of squares, the regression sum of squares, the bias fit coefficient c_0 , the r^2 correlation coefficient, the F value for the fit, and the significance level associated with that F value.

11.6.2 PERFORMING THE REGRESSION

The independent variables $\{x\}$ should be arranged in a matrix with the i th column containing the values for the i th term. The dependent variables $\{y\}$ should be arranged as a column vector or as a one-dimensional array.

Once the data is selected, a new screen appears listing the in-model and not-in-model variables and the overall fit statistics. A prompt list is located at the bottom of the screen for use in directing which variables are to be entered into or removed from the model. If you are entering variables, the regression routine will automatically select the best x_i to enter based on the smallest F-to-enter value. If you are removing variables from the model, the routine will automatically select the best x_i to remove based on the largest F-to-remove value. Of course, you may override these automatic choices and select any appropriate variable.

The current choice will be performed upon selecting **Continue** or by simply pressing the **Esc** key. Selecting **Quit** will terminate the regression process and place you in another prompt list that allows you to save the results in calculator Variables.

11.6.3 SAVING THE RESULTS

On quitting the **Regression** option, a menu of prompts appears in the upper-right corner of the display (refer to Screen R11.7). The prompts are used for saving the results of the analysis in calculator Variables. Specifically, the fit coefficients and their standard deviations can be saved along with the in-model variables and the covariance matrix of the in-model variables.

Stepwise Regression		Save Results	
Select independent var (R-Z): (X)		Fit coeff's (R-Z/None): (R)	
Start	End	Coeff st dev's (R-Z/None): (N)	
col: (1)	(5)	In-model x's (R-Z/None): (S)	
Select dependent var (R-Z): (Y)		In-model cov mat (R-Z/None): (T)	
Column: (1)		Press Esc to continue ...	
Continue/Quit/New Parms: * C *			
Menu Options		Variables	
Edit	Basic Stats	R= .0000	
Sort	Percentiles	S= .0000	
Hyp Tests	Histogram	T= .0000	
Distrib's	ANOVA	U= .0000	
Correlation	Regression	V= .0000	
Return		W= .0000	
		X= 8 x 5 REAL	
		Y= 8 x 1 REAL	
		Z= 4 x 8 REAL	

SCREEN R11.7

For each prompt below, the default response, No, will not save the corresponding results.

Fit coeff's

Enter the Variable letter (R-Z) in which to save the fit coefficients of the in-model terms (xi's). The resultant array will be a vector with length equal to one plus the number of terms in the model. Note that the fit coefficients are saved in sequence of increasing indices:

[c0 , c1 , c2 , ... , ci , ...]

with ci being the fit coefficient for the ith in-model variable.

Coeff st dev's

Enter the Variable letter (R-Z) in which to save the standard deviation of each fit coefficient. The resultant array will be a vector of the same length as the vector of **Fit coeff's**.

In-model x's

Enter the Variable letter (R-Z) in which to save the xi arrays that appear as terms in the model. The resultant array has dimensions $n \times m$, where n is the number of entries in each term and m is the number of terms in the model. The ith column in the matrix corresponds to the ith term in the model.

In-model cov mat

Enter the Variable letter (R-Z) in which to save the covariance matrix of the in-model terms. The matrix will be symmetric and of $n \times n$ dimensions, where n is the number of terms in the model.

References

- [1] R. J. Larsen and M. L. Marx. *An Introduction to Mathematical Statistics and Its Applications*, 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, 1986 .
- [2] J. D. Gibbons. *Nonparametrical Statistical Inference*, 2nd ed. New York: Marcel Dekker, 1985 .
- [3] S. James Press. *Applied Multivariate Analysis*. New York: Holt, Rinehart and Winston, 1972 .
- [4] K. Enslein, A. Ralston, and H. S. Wilf. *Statistical Methods for Digital Computers*, vol. 3. New York: John Wiley, 1977 .

CHAPTER 12: DIFFERENTIAL EQUATIONS

Differential Equations

Dimension of system (1 to 5): Continue/Quit: * C *

Equations

$$\begin{aligned} dY/dX &= 1 / 1+B*(Y-C)^2 \\ dZ/dX &= -A*(Z-C)^2 / (1+B*(Z-C)^2)^2 \\ dU/dX &= 2*A*(U-C) / (1+B*(U-C)^2)^2 \\ dV/dX &= 1 / 1+B*(V-C)^2 \\ dW/dX &= -A*(W-C)^2 / (1+B*(Z-C)^2)^2 \end{aligned}$$

Initial Conditions	Extrapolation Parameters
X = 1.0000	Final X = 10.0000
Y = .0000	Step size = .5000
Z = .0000	
U = .0000	Solve/Solve&Plot/New Eqs/Quit: * Q *
V = 1.0000	Variables X,Y,Z,U,V,W will be overwritten
W = 1.0000	
	Press Esc to continue

SCREEN R12.1

12.1 Overview

12.1.1 GENERAL DESCRIPTION

The **Diff Eqs** option generates solutions to either a single first order ordinary differential equation (ODE), or to systems of up to five first order ODE's. The fourth order Runge-Kutta method is used. The X Variable is always used to store solutions for the independent Variable and, if only a single equation is involved, then Y is used as the dependent Variable. If more equations are selected, then the U, V, W, and Z Variables are used as needed.

Solutions can be plotted using the **Graphics** option in the Main Menu, but **Diff Eqs** will also produce fixed format plots. The **Diff Eqs** solution plots are restricted to two formats, however: one for single equation solutions, and the

other for solutions to systems. In the latter case only three plots are shown: Y vs. X, Z vs. X and Y vs. Z.

12.1.2 HIGHER ORDER EQUATIONS

It is useful to note that a single higher order ODE can be solved using a system of first order equations, where the dimension of the system is equal to the order of the original equation. Thus, **Diff Eqs** can also be used to solve up to fifth order ODE's. More information on the equivalence of system and higher order equations can be found in most differential equations text books.

12.2 Specifying the problem

In order to produce a numerical solution to the initial value problem, the equation(s), the initial conditions, and the extrapolation Parameters must be specified. **Diff Eqs** prompts you for this information in three steps, beginning with input of the equation(s).

12.2.1 SPECIFYING THE DIFFERENTIAL EQUATIONS

When **Diff Eqs** is executed the entire screen is cleared and the following prompts displayed:

Dimension of system (1-5):

Continue/Quit:

The dimension must be a number from 1 to 5. A response of 1 is used for the case of a single equation. The **Quit** option will return control to the Main Menu.

When **Continue** is selected, a window opens in the middle of the screen in which the equations are to be reentered for each of the derivatives of the dependent Variables with respect to X. Between one and five prompts will appear, depending on the dimension selected. Each derivative prompt is followed by an "=", and a flashing cursor is displayed by the first equation.

Each equation should be entered as an algebraic expression, omitting the "\" that is required in command lines (see section 2.7).

The following editing keys are active:

Ins -- Toggles between insert and overwrite modes.

Del -- Erases the character at the cursor position.

- Backspace** -- Erases the character immediately preceding the cursor position.
- Home** -- Positions the cursor at the first character of the first line.
- End** -- Positions the cursor at the first character of the last line.

When **Esc** is pressed, the equations are compiled as they appear on the screen. If an error is encountered it is reported on the display. The original prompts for the system dimension and **Continue/Quit** are then repeated: select **Continue** to re-edit the equations, or **Quit** to exit the Differential Equations menu.

12.2.2 SPECIFYING THE INITIAL CONDITIONS

The initial conditions for a first order problem consist of a starting value, x_0 , for the independent Variable, together with initial values $y(x_0)$, $z(x_0)$, etc. for each of the dependent Variables. These Parameters are entered in a prompt list that appears once the equations are successfully compiled. The **Esc** key accepts input values and continues to the Extrapolation Parameters prompt list.

12.2.3 SPECIFYING THE EXTRAPOLATION PARAMETERS

The solution curves are built step by step, starting at the initial conditions. At each step the x -value is incremented by a small amount and corresponding increments of the dependent Variables are extrapolated. The process continues until the x -Parameter attains a final value. Thus, to complete the solution specifications, you must select values for the x -step size, and for the final x value. These Parameters appear in a prompt list that is displayed immediately after the initial conditions have been prescribed.

At the end of this list there is an action prompt that offers the options: **Solve**, **Solve and Plot**, **New Eqs**, or **Quit**. A description of these operations is given below:

Solve/Solve&Plot/New Eqs/Quit

Generates the solution curves and stores them in the Variables then returns to the first prompt of the **Diff Eqs** menu.

Solve/Solve&Plot/New Eqs/Quit

Generates the solution curves and stores them in the Variables then draws a fixed format plot. The plot is displayed along with a submenu containing the options:

- To Plotter** -- Draws the solution curve on the plotter
- Save on disk** -- Copies the display to a disk file that can be recalled to the full screen window in
 Graphics
- Return** -- Returns to the first prompt of the **Diff Eqs** menu.

Solve/Solve&Plot/New Eqs/Quit

Allows selection of different Parameters and equations by returning to the first prompt of the **Diff Eqs** menu.

Solve/Solve&Plot/New Eqs/Quit

Returns to the Desk Calculator Main Menu without generating the solution curves.

APPENDIX A: **FILE I/O ROUTINES**

This Appendix addresses the file conversion and data transfer capabilities of the File I/O menu. In the first part, the conversion of ASCII files (used by BASIC) into ASYSTANT files is considered. The second part deals with the transfer of data from an ASYSTANT file into a Desk Calculator Variable. The last part explains conversion of ASYSTANT files into DIF files and how to read such a file into Lotus 1-2-3.

A.1 Converting ASCII files to ASYSTANT files

A.1.1 OVERVIEW OF ASCII FILE CONVERSION

ASCII files are widely used, perhaps mostly because of their support by BASIC. ASCII is also the standard format used by many text editors and word processors. The main advantage of ASCII is this universality: you can move data into other programs and examine and edit files easily using a standard text editor such as edlin (provided with DOS), Wordstar (in the non-document mode), or the notepad editor provided with ASYSTANT.

There are two significant disadvantages to using ASCII representation. First, disk I/O is slow. The computer must translate data from its internal, binary format to the character representation of ASCII. This processing overhead can be significant, especially in data acquisition applications where speed is important. Secondly, ASCII files typically take more room on the disk than the binary representation of the same data.

Because of these disadvantages, ASYSTANT normally stores data arrays in "binary" files, i.e., data storage is identical to the PC's internal representation in memory. This makes disk I/O as fast as the hardware allows and permits smaller files.

ASCII data files, such as those created by a BASIC program, must therefore be converted into an ASYSTANT file in order to read the data into an ASYSTANT Variable. The File I/O menu contains a file conversion option to let you do this. The example here will cover the conversion of ASCII to ASYSTANT. The opposite procedure (ASYSTANT to ASCII) can also be carried out, essentially by performing these steps in reverse.

The file that will be converted is "STOCK.ASC". It contains data that are used in Session 6 on statistics. If you wish, take a moment to look at the file. (Ctrl-N will bring up the notepad editor, and specify STOCK.ASC for the filename). Note that the numerical data in this file are interspersed with text, and that text is separated from the data by quotes. Numbers are themselves separated by commas. This is the standard BASIC output format. If ASYSTANT is to convert a file, strings must be surrounded by quotes, and numbers separated by a delimiting character (usually a comma).

TIP

If you have an ASCII file which does not meet these requirements, go into an ASCII editor and edit the file (the notepad editor will accept a file under 32K bytes in length and with no line exceeding 80 characters).

The ASYSTANT file conversion process converts only numerical data; all text enclosed within quotes is ignored. Consequently, textual "headings" can be placed in the file without corrupting the numerical data but they will be lost when converted to the ASYSTANT file format. Keep in mind that ASYSTANT subfiles are images of the ASYSTANT Variables and must be either one-dimensional or two-dimensional (rectangular).

The ASCII conversion routine can operate in two modes: unformatted and formatted.

Unformatted conversion should be used if the ASCII data are not meant to be arranged in a rectangular fashion. In unformatted conversion, ASYSTANT takes on the organization of the data in the file. Each line containing numeric data in the ASCII file will become a new row in the output ASYSTANT subfile. The number of columns is set to the number of data points on the line. In cases where there is an unequal number of points/line, the conversion routine will create a subfile with as many columns as the longest single line in the ASCII file. Shorter lines will be padded with zeros to fill out the rectangle.

Formatted conversion is useful with data files which the user wishes to format into a rectangular array (hence the name). The user specifies the dimensions for the target subfile. Numbers in the data file are then converted one at a time, and placed element-by-element into the target subfile. If the conversion routine runs out of numbers on one line before a row of the subfile is filled, it will continue reading subsequent lines, until the row is complete.

This example will use formatted conversion, since the data are entries into a rectangular table. Unformatted conversion is explained in detail in the Reference section -- although its set up is very similar to the process described here. In fact in this case, since "STOCK.ASC" is laid out in rectangular form, either type of conversion will produce the same result.

A.1.2 CONVERSION EXAMPLE

To begin conversion, select **File I/O** from the Main Menu and then **Convert File Types**. You will be placed in another menu of conversion options. Select the **ASCII > Asystant** option.

A prompt list will now allow you to select the input and output files. [Refer to Screen A1.1]. Enter **d:STOCK.ASC** for the input file, where "d:" is the drive containing the file. The file is provided on the Analysis Diskette. Enter **d:STOCK.DAT** for the output filename, or another name if you prefer.

Stack Contents	Parameters
(empty stack)	A= .0000 B= .0000 C= .0000 D= .0000 E= .0000 F= .0000 G= .0000 H= .0000 I= .0000
----- ASCII to Asystant File Conversion -----	
Input file: (C:STOCK.ASC)	
Output file: (C:STOCK.DAT)	
Subfile type: Integer, Real: (R) Precision : Single, Double: (S)	
Quit/Continue: * C *	
	Variables R= .0000 S= .0000 T= .0000 U= .0000 V= .0000 W= .0000 X= .0000 Y= .0000 Z= .0000

SCREEN A1.1

The next two prompts allow you to specify the number type and precision of the data to be placed in the ASYSTANT file. The data in STOCK.ASC are real and representable in single precision, so accept the default values for these prompts. Continue moves on to a second prompt list [Refer to Screen A.2].

Stack Contents	Parameters
(empty stack)	A= .0000 B= .0000 C= .0000 D= .0000 E= .0000 F= .0000 G= .0000 H= .0000 I= .0000
----- ASCII to Asystant File Conversion -----	
Formatted/Unformatted conversion: (F)	
Start conversion on entry: (1)	----- Variables -----
Number of cols (0 = all): (8)	R= .0000
Number of rows (0 = all): (4)	S= .0000
Delimiter character: (,)	T= .0000
Quit/Continue: * C * Two-dim output	U= .0000
	V= .0000
	W= .0000
	X= .0000
	Y= .0000
	Z= .0000

SCREEN A1.2

You can now choose formatted or unformatted conversion. The default is unformatted, but enter F to select formatted for this example.

The "entry" prompt allows you to skip over any number of datapoints in the file before beginning conversion on the specified entry. This is useful, for example, if there are several datasets in a file. You can skip over any number of datasets by specifying an offset into the file which corresponds to the beginning of the dataset you want. It can also be used to skip over numbers in a file header.

In this example, the data within STOCK.ASC contain two separate tables. We will convert them to consecutive subfiles of the output ASYSTANT file in two operations. In the first, we convert the first table. In the second, we will specify an offset into the input file to skip the first table and convert only the second.

To create the first subfile, conversion should begin on the first number in the file(i.e., entry 1). This is the default value, so move on to the next prompt.

You can now specify the number of columns in the output subfile array. If you choose the default value of 0, then all the numbers in the file will be placed in a single one-dimensional array. However, there are 32 data points in the first table, and they should be set up in 8 columns, so enter 8 here. Note that a new prompt has appeared that allows you to select the number of rows for the output subfile. (If the output subfile is selected to be one-dimensional, this prompt is superfluous.) Selecting 0 for the number of rows will cause all the numbers in the file to be placed into an array 8 columns wide, with as many rows as necessary. If the conversion ran out of numbers in the middle of the last row, the row would be padded with zeros to fill it out.

In the first subfile, we do not want to convert all the numbers in the ASCII file, only the 32 making up the first table. That implies 4 rows ($8 \times 4 = 32$), so enter 4 into this prompt. Finally, the numbers in the table are separated by commas, so we should use the default delimiting character ",".

Select Continue and the first table will be converted into an ASYSTANT subfile. If the output file exists, this subfile will be appended to the end. If the output file that you chose is new, then this subfile will become the first subfile.

After this conversion is completed, ASYSTANT will return to the first prompt list. Note the "Quit/Continue" prompt defaults to "Q". Usually, you would press "Esc" to return to the File I/O menu. However, we still need to convert the second table in STOCK.ASC.

All of the values entered previously in this first prompt list are acceptable. Therefore, press End to jump down to the last prompt, and enter Continue. Since the output file specification is the same as before, the output subfile will be appended to the file just created.

"Formatted" conversion is already selected, so move to the next prompt. We want to skip the first 32 numbers, which were converted into the first subfile, so we must start conversion on the 33rd entry. Enter 33 into the offset prompt.

The second table should be organized into 6 columns and will fill out 8 rows. Enter 6 into the "number of columns" prompt. Since this is the last table in the file, we can simply let ASYSTANT convert every number to the end of the file. Enter 8 into the "number of rows" prompt to convert to the end of the file (alternatively, you could have entered "0" which automatically converts all rows in the file). The delimiting character is still a comma, so select Continue to start the conversion. An 8x6 subfile will be appended to STOCK.DAT. Once completed, Esc will return you to the menu of File Conversion options. Select Return to move back to the File I/O Main menu.

A.2 Reading the Converted Data into

Desk Calculator Variables

The data in STOCK.DAT can now be read using the "File>Var Transfer" option of File I/O, just like any other ASYSTANT file. The instructions in this section specifically set up three Variables needed by Session 6 with the data just converted.

Select **File/Var Transfer**, and enter **d:STOCK.DAT** in the "File Name" prompt. You will see a summary of the contents of the file printed on the screen as follows:

Start#	Shape	#Repts
1	4 x 8	1
2	8 x 6	1

Select **Continue** to enter the data-set specification prompt list at the bottom of the screen.

The first subfile contains the data for Table 1 of Session 6 and this entire subfile will be placed in Variable Z. Enter **Z** into the "I/O Variable" prompt. Notice how the current contents of the I/O Variable are reported to the right of the prompt. Next, we will select the location of the data to be read. Note that the values in this prompt list default to describe the complete first subfile: the subfile number is 1, the number of rows is 0 (all), the start column is 1, and the number of columns is 8. Consequently, we can easily read the first subfile by moving to the last prompt (**End**) and selecting **Read**. Notice that after the data is read, the contents of the Z Variable have been updated to show the shape of the first subfile: 4 X 8.

The data in the second subfile will not be read into a single Variable; instead, the first column will be read into the Y Variable, and the last 5 will be read into the X Variable.

Hit **Home** to skip back to the first prompt and select **Y** as the "I/O Variable". Next enter **2** for the subfile and notice how the later prompt values are automatically changed to reflect the size and shape of subfile 2. However, since we wish to read only column one (and all of the rows) of this subfile, change the "number of columns" prompt to **1**. Move to the last prompt and select **Read**. Y now contains the 8 X 1 array.

Now we want to read all the rows in columns 2 through 6 and fill the X Variable. Move to the "I/O Variable" prompt and enter **X**. Change the "start column" prompt to **2**. The "number of columns" prompt automatically updates to "5"; you cannot read more columns than there are in the table. Select **Read** in the last prompt to place the appropriate 8 X 5 array of data in X. Press **Esc** to Quit from the File/VAR Transfer menu.

[If you are working through Session 6, you may wish to select to **Return** to the Main Menu and resume that session now.]

A.3 Reading ASYSTANT Data into Spreadsheets

In this section we will describe how to move data in ASYSTANT files into a spreadsheet program. Specific instructions are offered for reading the data into Lotus 1-2-3 (TM).

The ASYSTANT file must be converted to a DIF file if it is to be read into a spreadsheet. DIF (for Data Interchange Format) was originally developed by Software Arts (the developers of Visicalc) as a standard for data exchange in the late 1970s. It has been adopted by many spreadsheet and database products including Lotus 1-2-3. DIF files include a file header which specifies the organization of the data in the file.

The major advantage of DIF over ASCII is that data is always organized in a rectangular array and the header will always give the dimensions of that array. A DIF file can contain only one array of numbers at a time. Converting a DIF file to an ASYSTANT file is therefore trivially simple: you need specify only the input and output filenames, along with the type and precision of the output subfile. The DIF file header will tell ASYSTANT all of the remaining information it needs!

Performing the reverse the conversion is even simpler: besides the filenames, we need specify only the subfile we wish to convert (remember that a DIF file can contain only a single table). We will work through a ASYSTANT to DIF conversion example now and then read the transformed file into Lotus 1-2-3. (The instructions are given for Version 1A of 1-2-3.)

Let us convert the second subfile just saved in STOCK.DAT to a DIF file. Select **Convert File Type** from the File I/O menu, and **ASYSTANT > DIF**. Enter **d:STOCK.DAT** for the input filename, and **2** for the subfile. The output filename can be anything you want, but Lotus will recognize a DIF file only if it has the extension **.DIF**. So enter **d:STOCK.DIF** for the output file. Continue will actually cause the conversion to execute. You can now exit ASYSTANT. If you wish to save any of your current Variables, select **Save/Exit** from the Main menu. Otherwise, press **Ctrl-E** for a fast exit.

If you are working with a hard disk and Lotus is in a separate DOS subdirectory, you will need to copy the DIF file to the Lotus subdirectory. This must be the directory containing the Lotus program (LOTUS, if you followed standard installation procedures), *not* the worksheet subdirectory (123FILES). Assuming you have just exited ASYSTANT (and the ASYSTANT subdirectory is current), the copy command is:

```
C> copy stock.dif \lotus
```

Floppy diskette users have it easier. Insert the diskette containing the STOCK.DIF file in the work drive and continue with the instructions.

When you enter Lotus, be sure to enter through the File Access System, not directly into 1-2-3. Do this by loading Lotus, not "123". The Access Manager menu will appear on the screen.

Select the **Translate** option, and then **DIF to WKS** from the translation system menu. You will then be prompted to select a source disk drive. Hard-disk users should select C:, floppy users should designate the drive that contains the file **STOCK.DIF**. Lotus will respond with list of all of the **DIF** files on the disk and prompt you to select a file for processing. Indicate **STOCK.DIF** and select a destination disk drive. This will usually be the same as the source drive.

Finally, Lotus asks you to select a translation order: columnwise or rowwise. Columnwise is the default, and is how **ASYSTANT** writes out a file, so you can simply press **Enter**. Then type **Y** to actually execute the translation.

Lotus has now created a file called **STOCK.WKS** on the destination drive. Select **Quit**, and then **Y**, to get back to the Lotus Access System. If you are working on a floppy based system, you can now go directly into 1-2-3. If you have a hard-disk system and your worksheets are normally stored in a subdirectory **123FILES**, you'll need to exit to **DOS**. Select **Exit** and then **Y**. You'll need to copy **STOCK.WKS** over to the **123FILES** subdirectory from **DOS**:

```
C> copy stock.wks \123files
```

Now enter 123. You can now load **STOCK** the way you would any other Lotus worksheet.

APPENDIX B: **ERROR MESSAGES**

This appendix is provided for two purposes: to explain the most commonly occurring error messages and to provide a complete listing of each error by number. This way the textual portion can be readily referenced if the ASYSTANT.MSG file is not available on the System disk drive. (If ASYSTANT.MSG is missing, only the error number appears on the screen.)

Section E.1 lists in numerical order the most common error messages followed by a brief summary of possible causes of the error; Section E.2 lists all error messages.

B.1 Common ASYSTANT Error Messages

ASYSTANT provides a complete error trapping system. This section presents just a few of the more common ones (there are over 200 error messages in all!) that you are likely to encounter, along with their probable causes.

Non-System disk or disk error**Replace and strike any key when ready**

This is not actually an ASYSTANT error message. It is an error message from PC-DOS. You are attempting to boot (start your system) with a data diskette, or the ASYSTANT distribution diskette. You must boot your PC with a diskette containing the PC-DOS operating system. Please see the appropriate chapters in the IBM PC-DOS manual on creating a diskette with PC-DOS on it.

System restarted

Also not really an error message. This message, which appears after you type Control-Break, acknowledges that the ASYSTANT system has been restarted; that is, that the Control-Break recovered control of the system.

Error 5 -- name is undefined

You have typed a word at the command line which ASYSTANT does not recognize. The most common causes are misspelling or improper spacing of commands. Remember:

1. There must be at least one space between each operator if using RPN (Reverse Polish Notation).
2. Command words, such as **store**, cannot be used within algebraic expressions. They may only be combined with other words and operators in RPN.

Error 11 -- file not found

You have attempted to access a file that does not exist. You have either specified the wrong file name or the wrong disk drive, or you are using the wrong diskette. Use the **Directory** option in the menu of DOS Operations (**Ctrl-D**) to find out what is actually on the selected diskette.

Error 29 -- Insufficient disk space

You have attempted to create or write a file to a disk that does not have enough room to store the file. You will have to either store the new file on another disk or delete some of the files on the current disk. The **Directory** option in the menu of DOS Operations (**Ctrl-D**) can be used to determine how many bytes of the disk are already in use.

Error 83 -- not enough entries on the #stack

Most ASYSTANT words require a specific quantity of information to be on the stack when they are called. For example, the SWAP operator requires two entries on the stack to swap. Remember that in RPN, each operation is performed sequentially from left to right and acts on the top stack entry or entries.

Error 89 -- binary operations require two entries on the #stack

This is similar to Error 83, but is generated specifically by binary operations such as + or *.

Error 153 -- out of available memory, unable to create array

All arrays take up a certain amount of memory. ASYSTANT cannot reserve more memory for an array than is available to the system. You must reduce the size of the array, or use a number type that requires less memory (refer to Table R2.1 in Chapter 2). Alternatively, you may free memory currently allocated by:

1. Clearing the Variables (R-Z) by entering `clear.vars` at the command line (first save needed arrays on disk via the File I/O option in the Main Menu).
2. Clearing the stack by entering `clear` at the command line.

(See Chapter 2, section 2.10, for a description of these command words.)

Note: Often, the occurrence of Error 153 does not seem to be directly related to the operation being performed. ASYSTANT uses memory to store intermediate computations. If there is not enough memory, this message can occur midway through the operation.

Error 155 -- invalid 8087 operation

This is a generic error trapped by the 8087 chip. It indicates that the requested operation cannot be performed because the operands are unsuitable. For example this error is produced when attempting to take the square root of a negative real number.

B.2 Numerical Listing of ASYSTANT Error Messages

```
Error 0  -- unknown system error -- contact Macmillan Software
          Company
Error 5  -- name is undefined
Error 6  -- disk address out of range
Error 8  -- disk error
Error 10 -- file read error
Error 11 -- file not found
Error 12 -- a name must be specified
Error 14 -- missing decimal point
Error 15 -- ASYSTANT Version 1.00
Error 16 -- error while creating file
Error 17 -- compilation only, use in definition
Error 18 -- execution only
Error 20 -- definition not finished
Error 23 -- can't resume; earlier error altered system state
Error 27 -- duplicate filename or file not found
Error 28 -- Error while loading .OVL file
Error 29 -- Insufficient disk space
Error 30 -- File cannot be copied to itself
Error 32 -- file write error
Error 36 -- requires either a STRING or STRING ARRAY
Error 37 -- requires two STRINGs
Error 47 -- file not found: ASYSTANT.HLP
Error 50 -- requires a STRING
Error 53 -- illegal value encountered when displaying stack contents
Error 54 -- Can't find master diskette - exiting ASYSTANT
Error 55 -- System integrity error - exiting ASYSTANT
Error 56 -- System does not have properly configured 8087 installed
Error 57 -- Type any key to continue...
Error 58 -- COPY or RENAME require unambiguous filenames
Error 59 -- name is undefined
Error 62 -- Master diskette not authorized for DATA ACQUISITION
          module
Error 63 -- Not enough memory to execute ASYSTANT
Error 66 -- Out of memory when aligning DMA arrays
Error 70 -- integer divide by zero
Error 71 -- parity check error
Error 72 -- NMI interrupt
Error 79 -- Overlay is not available on the overlay drive
Error 80 -- unknown math error -- contact Macmillan Software
          Company
Error 81 -- not enough space for unnamed array
Error 82 -- can't reference a non-positive stack entry index
Error 83 -- not enough entries on the #stack
Error 85 -- not enough entries on the #stack
Error 86 -- not enough room for number on the #stack
Error 87 -- not enough numbers on the #stack
```

- Error 88 -- Operation requires arrays with an equal number of dimensions
- Error 89 -- binary operations require two entries on the #stack
- Error 90 -- unary operations require one entry on the #stack
- Error 91 -- real>complex conversions require a real entry on the #stack
- Error 92 -- complex>real conversions require a complex entry on the #stack
- Error 93 -- operation requires a real or integer entry on the #stack
- Error 94 -- comparisons require two real or integer numbers on the #stack
- Error 95 -- illegal origin for SUB dimension
- Error 96 -- array index out of bounds
- Error 97 -- no array setup for operation
- Error 98 -- storing data into a scalar/array requires an entry on the #stack
- Error 99 -- attempt to store a non-scalar number into a scalar
- Error 100 -- type mismatch when storing into a scalar/array
- Error 101 -- array dimension size must be greater than zero
- Error 102 -- array not specified when accessing array element via [...]
- Error 103 -- illegal number of indices when accessing array element via [...]
- Error 104 -- too many parameters for SUB dimension
- Error 105 -- array not specified for SUB
- Error 106 -- illegal number of indices for SUB
- Error 107 -- an array must be specified
- Error 108 -- array not specified for TRANS
- Error 111 -- index out of range
- Error 114 -- illegal number of indices for XSECT or DIAG
- Error 117 -- UNNAMED.ARRAY dimension size must be greater than zero
- Error 120 -- operation requires an entry on the #stack
- Error 121 -- FFT requires an array on the #stack
- Error 123 -- graphics device not initialized
- Error 126 -- graphics auto plotting requires a single 1 dim array
- Error 127 -- graphics auto plotting requires a two identical 1 dim arrays
- Error 128 -- graphics routine not applicable to current graphics device
- Error 129 -- unfinished algebraic expression
- Error 130 -- unfinished algebraic expression on left hand side of =
- Error 131 -- graphics array readout array must be 1-dimensional
- Error 132 -- graphics array readout array cannot be unnamed
- Error 133 -- logarithmic autoplotting can only be done for positive values
- Error 135 -- too many unnamed arrays on stack
- Error 136 -- unnamed array bigger than 64k bytes
- Error 137 -- array dimensions are incompatible
- Error 138 -- trying to assign an array into a scalar
- Error 139 -- operation requires an array
- Error 140 -- parameter must be real or integer

Error 141 -- parameter list syntax error
Error 142 -- determinant is zero, can't solve system of equations
Error 145 -- specified array size exceeds 64k bytes
Error 146 -- arguments for CATENATE must have an equal number of dimension
Error 148 -- incompatible array dimensions for MATRIX*
Error 149 -- KRONEC* arguments must be arrays
Error 150 -- MATRIX* arguments must be arrays
Error 151 -- LOOKUP arguments must be arrays
Error 152 -- stored unnamed array no longer available due to CLEAR
Error 153 -- out of available memory, unable to create array
Error 154 -- parity check error from memory or I/O device
Error 155 -- invalid 8087 operation
Error 156 -- denormalized 8087 operand encountered
Error 157 -- divide by zero
Error 158 -- numerical overflow
Error 159 -- numerical underflow
Error 160 -- unknown 8087 error condition
Error 161 -- cannot autoplot a COMPLEX array
Error 162 -- numerical output error during STACK DISPLAY; stack cleared
Error 163 -- no more arrays currently defined
Error 164 -- incorrect index for number list expression
Error 165 -- a named array must be specified
Error 166 -- dims specified for RESHAPE are not compatible with input array
Error 167 -- the input argument for SORT or INDEX must be an array
Error 168 -- complex arrays cannot be sorted or indexed
Error 169 -- only 1-dimensional arrays can be used
Error 170 -- the degrees of freedom for the chi-squared dist must be > 0
Error 171 -- the degrees of freedom for the student-t dist must be > 0
Error 172 -- can't take LN or LOG of a negative, non-complex number
Error 173 -- assigning into a number is not allowed
Error 174 -- Can only execute when current graphics device is IBM GRAPHICS
Error 175 -- cannot autoplot an array with constant values
Error 175 -- KRONEC* products require 1 or 2 dimensional array arguments
Error 178 -- illegal syntax for algebraic expression
Error 187 -- AEDIT requires a one or two dimensional array to edit
Error 188 -- SYNC.PERIOD can't be 0
Error 189 -- too much time elapsed before SYNCHRONIZE
Error 192 -- can't EDIT with multitasking active
Error 198 -- STRIP.CHART data range must be non-zero
Error 199 -- strip chart data array must be 1 or 2 dimensional
Error 200 -- can't evaluate gamma function at non-positive integers
Error 201 -- can't evaluate Bessel functions with complex order

- Error 202 -- scroll array must be 1 dimensional
- Error 203 -- need a bigger array to scroll
- Error 205 -- Can't access disk file
- Error 207 -- Operation requires a scalar
- Error 223 -- Illegal character in file name
- Error 256 -- Conditional is never satisfied
- Error 257 -- Function definition is too long -- the system may be corrupted
- Error 272 -- unknown DAS error -- contact Macmillan Software Company
- Error 279 -- function not valid for current DAS device
- Error 282 -- DAS device not initialized
- Error 289 -- a DIGITAL MASK value cannot be 0
- Error 291 -- too much time elapsed before SYNCHRONIZE
- Error 295 -- TASK PERIOD too large (must be less than 53 msec.)
- Error 296 -- TASK PERIOD too small
- Error 305 -- invalid digital mode for current 8255 operating mode
- Error 306 -- invalid digital mode for current DAS device
- Error 307 -- DAS device not configured in this version of ASYSTANT+
- Error 308 -- can't execute while tasking is active
- Error 317 -- Unknown hardware error code
- Error 318 -- DAS command overwrite error
- Error 319 -- Conversion delay time too small
- Error 320 -- Illegal digital mode for DAS device
- Error 323 -- Internal or external clock too fast for block mode D/A output
- Error 327 -- Internal or external clock too fast for block mode A/D input
- Error 328 -- Internal or external clock too fast for block mode A/D input
- Error 333 -- Current device does not use 8255 for digital I/O
- Error 334 -- Conversion delay time too large
- Error 337 -- there are no DAS devices presently configured
- Error 344 -- current DAS device must be SERIES.500
- Error 346 -- CONVERSION DELAY must be larger than SERIES 500 SETTLING TIME
- Error 352 -- multitasking interrupt overrun condition
- Error 359 -- specified A/D channel# does not have a CJR associated with it
- Error 360 -- specified A/D channel# must be on an AIM6 board
- Error 361 -- WFS.EXE file has not been loaded
- Error 368 -- unknown analysis error -- contact Macmillan Software Company
- Error 369 -- operation requires an input argument on the top of stack
- Error 370 -- a number was used for an input argument that required an array
- Error 375 -- SOLVE.EQS requires two arguments on the number stack
- Error 377 -- the coeff array input to SOLVE.EQS was not 2-dimensional

Error 378 -- the two arguments input to SOLVE.EQS had incompatible
dims
Error 379 -- the input array was not 2-dimensional
Error 380 -- a degenete matrix was encountered
Error 381 -- non-zero constant polynomials have no roots
Error 382 -- differentiation order cannot exceed 7
Error 383 -- differentiation order cannot be less than 1
Error 384 -- differentiation degree cannot exceed 7
Error 385 -- differentiation degree cannot be less than 1
Error 386 -- differentiation degree cannot be less than the order
Error 389 -- SMOOTH cutoff frequency cannot exceed .5
Error 390 -- SMOOTH cutoff frequency cannot be less than .03
Error 392 -- illegal syntax for CURVE FIT
Error 400 -- New text file
Error 401 -- File too large for buffer
Error 402 -- Cannot read file
Error 403 -- Not enough disk space for saving file
Error 404 -- Cannot close file
Error 405 -- No end-of-file marker in file
Error 406 -- Text lines must be less than 80 characters
Error 407 -- File changed since save
Error 408 -- No more memory
Error 409 -- Two marks are not defined
Error 410 -- Block too big to move or copy
Error 411 -- File changed since block marked
Error 412 -- Cursor at end of file
Error 413 -- End mark must be past beginning mark
Error 414 -- Unknown string operation to repeat
Error 415 -- Saving file, please wait...
Error 416 -- Cannot delete current backup file
Error 417 -- Cannot create file with saved text
Error 418 -- Cannot rename backup file
Error 419 -- Type file specification or <cr>:
Error 420 -- Type directory specification of <cr>:
Error 421 -- Edit at last error? (Y/N)
Error 422 -- Reading file, please wait...
Error 423 -- Checking file, please wait...
Error 424 -- Illegal filename for edit
Error 425 -- Save changed file? (Y/N)
Error 426 -- Replacement string too long
Error 427 -- End of line will be lost
Error 432 -- Diskette is write protected
Error 433 -- Unknown unit
Error 434 -- Drive not ready
Error 435 -- Unknown command
Error 436 -- Data error
Error 437 -- Bad request structure length
Error 438 -- Seek error
Error 439 -- Unknown media type
Error 440 -- Sector not found

Error 441	--	Printer out of paper
Error 442	--	Write fault
Error 443	--	Read fault
Error 444	--	General DOS failure
Error 512	--	subfile specification out of range
Error 513	--	subfile length greater than 64K bytes
Error 517	--	too many subfiles in file
Error 520	--	Requires an array to be specified
Error 521	--	specified data file template is too large
Error 522	--	invalid data file template
Error 523	--	data file must be opened
Error 524	--	data file read error
Error 525	--	data file write error
Error 526	--	data file close error
Error 527	--	current data file not closed
Error 528	--	data file open error -- file not found
Error 529	--	data file comment line out of specified range
Error 530	--	data file write error when creating file
Error 531	--	data file create error
Error 532	--	BASIC data file is not open
Error 533	--	BASIC data file is already opened
Error 534	--	BASIC data file not found
Error 536	--	requires an array entry on the #stack
Error 537	--	data file access error - array not compatible with subfile
Error 538	--	attempt to access invalid data file
Error 539	--	can't append array to file; file incompatible with ASYSTANT
Error 540	--	random file create error
Error 541	--	random file close error
Error 542	--	random file seek error
Error 543	--	random file open error
Error 544	--	random file write error
Error 545	--	random file read error

INDEX

Menu options, parameters, and commands are listed in **boldface**.

datasets: T5-5, R8-9
of columns: T5-5, R7-8
* R8-19
+ R8-19
- R8-19
.\$\$\$ files R1-20
/ R8-19
1 Sample Chi-sq test R11-7
1 Sample t test R11-7
1 way ANOVA R11-12
10^ R8-15
10exp R2-16
2 Sample Chi-sq test R11-7
2 Sample t test T6-9, R11-7
2 way ANOVA R11-12
2d:fft R2-28
2d:ifft R2-29
<>n:search R2-22, R2-33
<n:search R2-22, R2-36
= T7-3, R2-10
=n:search R2-22, R2-33
>mem T4-6, R2-36
>n:search R2-22, R2-33
? T1-25
?Endpts R6-12
Abbreviations for keys T1-5
Abs R8-15
abs R2-15
acos R2-15
acosh R2-15
Add R10-2
aedit R2-18 thru 19
Algebraic expressions T1-14
Amplitude Compression Factor R4-5
Analysis T5-16, R6-6
Analysis Diskette files T1-2
Analysis of Variance (see ANOVA)
Angular units R2-13
ANOVA T6-7, R11-5
ANOVA and hypothesis testing T6-3 thru 7
ANOVAs, types R11-2
Append R8-11 thru 12
Appending data R7-5
Arithmetic T1-13 thru 15, R2-6 thru 7

Array arithmetic T2-2 thru 4, R2-9 thru 10
Array editing R2-18 thru 19
Array Operations T7-8, R2-2
Array Operations menu R2-17 thru 22
Arrays R2-8 thru 10
Arrays and Graphics T2-1 thru 13
ASCII > ASYSTANT R7-16
ASCII file conversion R7-13 thru 15
ASCII files format R7-2
ASCII to ASYSTANT file conversion A-1 thru 5
asin R2-14
asinh R2-14
ASYSTANT/ASYSTANT+, differences I-1
ASYSTANT > ASCII R7-18
ASYSTANT > DIF R7-16
ASYSTANT data files structure R7-3 thru 6
ASYSTANT errors R1-25
ASYSTANT file conversion R7-16
ASYSTANT program diskette files T1-2
ASYSTANT system configuration R1-6
ASYSTANT.CFG file T1-11
ASYSTANT.HLP R1-11
ASYSTANT.MSG R1-11
atan R2-15
atanh R2-15
Auto/Set R6-12
autocorr R2-30, R8-17
avg R2-31
Axis (Y or N) R4-7
Axis parameters R6-11
Axon T7-5, R4-5
Axonometric plots T7-5 thru 8
BFGS method R9-12
BIOS ROM T1-27
B-window R2-35
Backspace key R3-3
Basic Stats T6-8, R11-4
Bessel function T7-2, R2-25 thru 26
Betax(a,b) R2-26
Binary Ops T5-7
Binary operations R8-19
Blackman window R2-30 thru 35
Block oper R1-21
bye R2-36
Calculator Functions T1-12, R2-2
Calculator Functions Menu R2-12 thru 16
Cartesian coordinates, conversion R2-24 thru 25
Cartesian plots R4-2
cat R2-20
Catenate R8-20
Chi Sq %'s R11-9
Chi Sq %tiles R11-10
Clear R4-9
clear R2-36

Clear M R5-5
clear.vars R2-36
Clip R6-14
Coeff st dev's R11-17
Coefficients R9-4
Coeffs>Var R10-2
Color configurations R4-9
Color on: T1-7 thru 8
Color T2-7
Colors T7-6, R4-9
Command line R2-2, R2-36
Comments R7-3
Comments, editing R7-6
Complex numbers, entering R2-4 thru 5
Configuration files R1-10
Configuration menu T1-6
Configuration, hard-disk R1-12
Configuration, two-floppy R1-12
Configuring a plotter R1-8
Configuring a printer R1-8
conj R2-23
Contour T7-7, R4-6
Contour plots T7-5 thru 8
Conversion of files R7-2
Conversions and special functions R2-2, R2-23 thru 26
Convert file types R7-2, R7-12 thru 18
Converting ASCII files to ASYSTANT R7-13 thru 15
Converting ASYSTANT files to DIF and ASCII R7-16
Converting DIF files to ASYSTANT R7-15
convolve R2-30, R8-20
Copy file R1-23
Correlation R11-5
Correlation matrix R9-4
cos R2-15
cosh R2-15
Cosine R5-7
crosscorr R2-30, R8-20
Ctrl-Break R1-4
Ctrl-D T1-24, R1-4
Ctrl-F R1-4
Ctrl-E T2-13, R1-4
Ctrl-N T1-19, R1-4
Ctrl-P R1-4
Ctrl-V T2-11, R1-4
Current DS R8-3
Current segment analysis R6-15
Current segment processing R6-13
Current segment, processing and analyzing R6-3
Current segment: T5-15
Curve Fit T3-5, R9-1
Curve area R6-16
Curve fit options R9-5 thru 10
Curve fitting R9-1 thru 14
Curve fitting algorithms R9-11

Custom overlay disks T1-11
Data flow during processing R8-5
Data processing R8-2 thru 3
Data transfer R7-5
Data transfer between files and variables R7-1
Data type R2-13
Dataset Location R8-3
Dataset definition R7-4
Dataset processing R8-4
Dataset selection R7-4
Defining user functions T1-18
deg R2-34
Del key R3-3
Delete File T1-24, R1-23
Deleting text R1-19
Delimiter character R7-17
Derivative R6-14
Desk Calculator T1-1 thru 27, R1-2, R2-1 thru 36
Desk Calculator variables, reading data into A-6 thru 6
Desk Calculator windows T1-12, R2-2
Desk Calculator menus R1-2, R2-2
Desk Calculator, command line editing R2-11
Desk Calculator, complex numbers R2-4
Desk Calculator, entering numbers R2-4 thru 5
Desk Calculator, function keys R2-11
determ R2-32
diag R2-20
Diagnostics for fitting R9-3
DIF > ASYSTANT R7-16
DIF file conversion R7-15
DIF files format R7-2
diff R2-28, R2-33
Diff Eqs T3-2, R12-2
Differential equations R12-1 thru 4
Differentiate R10-3
Directory R1-24
Disk file setup R1-9
Display T2-13, R4-8
Display Graph T2-6, R4-3
Display hardware options R1-7
Display hardware specification R1-6
Display key T7-7
Distrib's R11-5
Distrib's menu R11-9
Div R10-2
DOS operations T1-24, R1-23
drop R2-14
DS Average T5-7, R8-17
DS Cat & Smooth R8-17
DS Reshape R8-18
DS Std'd Dev R8-18
DS Subsection R8-18
dup R2-14
dY/dX= T3-3

E(m) R2-26
Edit R11-3
Edit menu R1-16
Editing data file comments R7-6
Editor R5-11
Editor restrictions R1-16
Editor, exiting R1-21
Editor, function keys R1-17 thru 19
Editor, status line R1-19
EGA card T2-7, T7-6
EKG waveforms T5-1 thru 18
elliptical integral R2-26
End key R3-3
end.window R2-36
Envelope R6-15
Erase R4-8
Erase Graph R4-3
erf R2-25
Err² R9-2
ERR.TMP R1-25
Error handling R1-25
Error messages R1-11, T1-26, B-1 thru 9
Esc key R3-3
Exchange T5-14, R6-18
Exiting an edit session R1-21
Exiting ASYSTANT T1-27
exp R2-15
Exponential R9-7
Extrapolation parameters R12-3 thru 4
F %'s R11-10
F %tiles R11-10
F-value T6-13
fft R2-28, R8-16
Fall time R6-16
File T1-10
File > File T5-2, R8-2
File > Var R8-2
File copying R1-23
File deletion R1-23
File I/O T5-17
File I/O routines A-1 thru 8
File Input/Output R7-1 thru 18
File Input/Output T5-17
File Name R7-7
File Proc T5-2, R8-1
File Processor R8-1 thru 22
File renaming R1-23
File type conversion R7-2
File types R1-12
File-based processing T5-2
File/Var Transfer R7-7
Files, help R1-11
Files, main system and configuration R1-10
Files, overlay R1-11

Files, saving and restoring R1-4
Files, user R1-12
Files, work R1-11
First input variable R8-21
fit R9-9, T3-9
Fit coeff's R11-17
Fit Tolerance R9-9
Fit type R9-9
Fitted values R9-3
Formatted conversion R7-14
Formatted/Unformatted R7-17
Fourier transformation R2-28 thru 29
Full T2-12
Function editing R3-3
Function keys R1-22
Functions T2-8, R3-1
Functions, arguments R3-2
Functions, nesting R3-2
Gamma R2-26
Gauss-Newton method R9-12
Generate R5-4, R11-5
Goodness of fit T6-5, R9-2
grad R2-34
Graphics R4-1 thru 10
Graphics T2-5
Graphics menu T2-5, R4-3 thru 10
Grid (Y or N) R4-7
Hard-disk system setup T1-3
Help T1-25, R1-14 thru 15
Help Diskette files T1-2
Help file R1-11
Help file, drive T1-10
Hermite (polynomial) R10-3
Hilbt (Hilbert transformation) R2-30
Histogram R11-4
Home key R3-3
Horizontal Shift R4-5
Horizontal Specifications R5-9
Horz Specs T4-4, R5-9
Hyp Tests T6-4, R11-4
Hyp Tests menu R11-6
Hypothesis testing T6-2 thru 7, R11-6
I/O Status R8-3
I/O Variable (R-Z) T5-17, R7-8
IBM Enhanced Graphics (see EGA card)
ifft R2-28
In model T6-12
In-model cov mat R11-18
In-model x's R11-17
index R2-22
Index/World T5-11, R6-12
Inplace R8-11 thru 12
Input File T5-3
Input and output variable selection R8-21

Input file specification R8-7 thru 10
Input specification R8-5
Input variable R8-21
Input variable selection T5-8
Ins key R3-3
integ R2-28
Integral R6-14
Integrate R10-3
inv R2-14
inv\mat R2-32
Invert R5-11
Jn R2-25
K(m) R2-26
KS Normality test T6-5, R11-6
Key Diskette files T1-2
Key abbreviations T1-5
Keys, combination R1-22
Keys, function R1-22
Keys, special R1-4
Kolmogorov-Smirnov normality test R11-6
kronec* (Kronecker product) R2-32
Label T2-12, R4-8
Labeling plots T2-12 thru 13
labels R2-24
Labels (Y or N) R4-7
Laguerre (polynomial) R10-3
lam R2-20
Laminate R8-20
Left Half T7-5
Legendre (polynomial) R10-3
Levenberg-type modification R9-13
Line editing T1-17
Linear R9-5
Lissajous figures T2-8
List R7-11
ln R2-16
Locate R4-8
Log R8-15
Log (Y or N) R4-7
Logarithmic R9-6
Locate T2-13
lookup T7-8, R2-22
M=M+O T4-5, R5-6
Main Menu T1-12
Manip M R5-3
Manip O R5-3
Mann-Whitney Rank-sum test R11-8
matrix* R2-32
Matrix operations R2-27 thru 35
Max R4-7
max R2-31
Max Length T5-4
Maximum Iterations R9-9
MEM T4-6, R2-36, R6-2

Mem ovl T5-14, R6-10
Memory Ops R6-6
Memory allocation R1-4
Memory operations R6-4, 17
Memory ops R6-17
Memory waveforms R5-2
Menus and prompt lists R1-2
min R2-31
Min R4-7
Min/Max R5-11
Mini Calculator T4-6
Mini Calculator T2-10 thru 11, R1-24
Mini Calculator and memory segment R6-4
mod R2-23
MODE command (DOS) R1-8
Moving blocks of text R1-20
Mult R10-2
Multilinear R9-8
Multilinear function fitting R9-2
n! R2-25
n:C:r R2-25
n:P:r R2-25
n:ramp R2-8, R2-20, R2-33
n:reshape R2-22, R2-34
n:rot R2-20
n:search T7-8, R2-22, R2-33
neg R2-15
New Eqs R12-4
New Label T2-13, R4-8
New Parm's R4-3
New file R7-11
Newton-Raphson method R9-12
next R2-12
nm:ramp R2-20, R2-33
nm:reshape R2-22, R2-34
Nesting of functions R3-2
Noise waveforms R5-3
Normal %'s R11-9
Normal %tiles R11-9
Not in model T6-12
Note #'s R6-17
Notepad T1-19 thru 23, R1-16 thru 22
Notepad, creating a table T1-21 thru 3
Notepad, entering numbers T1-21
Nrml Noise R5-8
Number of cols (0=all) R7-17
Number of rows (0=all) R7-17
Number types R2-5, R2-13
Numeric Output Format R2-13
Numeric keypad keys R5-12, R6-7 thru 9
Offset (columns) R8-8
ORG R6-2
Orig R6-9
Origin (B/M/T) (R/M/L) R4-7

Other Specs T4-6, R5-9
Output file selection T5-6
Output file specification R8-11 thru 13
Output specification R8-5
Output variable R8-22
Output variable selection T5-8
Output waveforms R5-2
Overlay files R1-11
OVL-OFF T5-14
OVL-ON T5-14
Ovl tgl R6-10
Ovl tgl T5-14
Ovrwrt R6-9
Parameters T1-12
Parameters and Variables T1-16, R2-3
Parm Display T7-6, R4-6
Part Corr T6-13
Partial correlation T6-13
PC compatibles T1-27
Peak width R6-16
pedit T1-16, R2-13
Pendulum motion T3-1 thru 9
Percentiles R11-4
PgDn key R3-3
PgUp key R3-3
pi R2-14
Plot R7-11
Plot M R5-6
Plot M+O T4-6, R5-6
Plot O R5-6
Plot ParmS T5-14
Plot Wfrm T4-2 thru 3, R5-5
Plot data T5-14, R6-10
Plot parms T5-11, R6-6
Plot tgl R6-10
Plots, simple T2-5
Plotter R11-5
Plotter, configuring R1-8
Plotting data T5-11
Plotting options R6-10
Plotting parameters R6-3
pol>xy R2-24
Polar coordinates, conversion R2-24
Poly Coefficients R9-4
Poly scale R6-14
Polynomial T3-6, R9-6
Polynomial display R10-1
Polynomial operations R2-37, R10-1 thru 4
Polynomial operations menu R10-2
Power spectra R2-29, R6-6
Prediction by regression T6-10 thru 15
Prev R6-9
Prev ovl R6-10
print R2-35

Printer, configuring R1-8
Process T5-12, R6-6
Processing a segment of a waveform T5-12
Processing operations R8-14 thru 20
Processing procedures R8-6
PRV T5-13, R6-3
Pssn Pulse (Poisson Pulse) T4-5, R5-8
Pulse R5-7
Pwrspec R2-29, R6-14
Quit R7-11
Quit&Save R5-4
Quit/Continue R7-8
R^2 R9-2
ra>z R2-24
rad R2-34
Read R7-9
Recall R4-10
Recall/Setup/Continue R1-10
Reconfigure R1-8
Regression T6-11, R11-5
Regression SS T6-13
Regression analysis T6-10 thru 15
Rename file R1-23
reset R2-36
reshape R2-22, R2-34
Residual SS T6-13
Residuals R9-3
Reverse R5-11
reverse R2-20
Reverse Polish Notation T1-14, R2-6 thru 7
Right Half T2-9
Rise time R6-16
roll R2-14
Roots R10-2
Rotate R5-11
Row (0=all) R7-8
Row Dimensions T7-4
RPN (see Reverse Polish Notation)
Runge-Kutta algorithm T3-4
Save R4-10
Save fitted values T3-6
Save on Disk R12-4
Save/Exit T1-27
Saving and restoring files R1-4
Saving modified files R1-20
Sawtooth R5-7
Screen layout R8-3
Scroll R6-10
Scroll and mark R6-7
Second input variable R8-22
SEG T5-13, R6-3
Seg Mem R6-18
Seg + Mem R6-17
Seg - Mem T5-14, R6-17

Seg / Mem R6-18
Seg > Wfrm R6-9
Segment analysis T5-16
Segment manipulation R6-7 thru 9
Segment, processing a waveform T5-12
Selection of input and output variables R8-21 thru 22
Set axes T5-11, R6-6
Set horiz T5-11, R6-11
Set vert R6-12
set.degree R2-37
set.frequency R2-37
set.order R2-37
Setting up disk files R1-9
Setup T2-7, R4-7
Setup instructions T1-3
Setup menu R1-6
Shift R10-2
show:save T3-6, R9-10
Sig R9-2
sin R2-14
sinh R2-14
Sine T4-2, R5-7
Sinusoid T4-9
sJn R2-26
Smooth T5-13, R2-27, R2-34
Solve R12-3
Solve&Plot R12-4
solve:eqs R2-31
sort R2-22, R11-4
Space T2-10
Special keys R8-15
sph>xyz R2-25
Spherical coordinates, conversion R2-25
Spreadsheets, reading ASYSTANT data into A-7 thru 8
Sqrt R2-16, R8-15
Square T4-3, R5-7
Stack R2-3 thru 7
Stack Contents T1-12
Standard Deviations R9-4
Start Column: T5-17, R7-8
Start conversion on entry R7-17
start.window R2-37
Statistics T6-8 thru 9, R11-1 thru 18
Statistics menu R11-3 thru 5
Stats T5-16, R11-1 thru 2
status T2-8, T4-9, R2-13, R2-34
stdev R2-31
stepwise regression R11-2
Stepwise regression R11-14 thru 18
Stepwise regression results R11-17 thru 18
Stock portfolios T6-1 thru 15
store R2-12
Student t %'s R11-10
Sub (polynomial subtraction) R10-2

sub (subarray extraction) R2-18, R2-35
Subarray, definition R2-8
Subfile: T5-17, R7-8
Subfile blocks R7-3
Subfiles R7-3
swap R2-14
sYn R2-26
System Diskette files T1-2
System configuration R1-6
System files R1-10
System requirements I-3
System status modification R2-13
Tab T4-2
Table ANOVA R11-13
tan R2-15
tanh R2-15
Tchebyshev (polynomial) R10-3
Technical support I-4
Temporary work files, drive T1-10
Text deletion R1-19
Text in ASCII and DIF files R7-12
Text search and replace R1-20
Theoretical distributions R11-9
To Plotter R4-3
Tol T6-13
Tolerance T6-13
trace R2-31
trans R2-18
Triang R5-7
Two-dimensional plots T7-2 thru 4
Two-dimensional representations R4-2
Two-floppy system setup T1-4
Unary Ops T5-7
Unary operations R8-15
Unformatted conversion R7-14
Unif Noise T4-2, R5-8
User files R1-12
User Function T3-7, R9-8
User function fits, trouble shooting R9-13
User function fits R9-11 thru 14
User function operations R8-14
User functions, defining T1-18, R2-3, R2-37, R3-1 thru 3
Var > File R8-2
Var > Coeffs R10-4
Variable selection R8-21
Variable x T6-12
Variables T1-12
Variables T1-16
Variables, transferring to disk T5-16
Vert Specs T4-3, R5-9
Vertical Base Fraction R4-5
Vertical Specifications R5-9
Vuport Color T7-6
Wave and matrix operations R2-2

Wave Gen T4-2
Wave generator T4-1 thru 8
Wave operations R2-27 thru 35
Waveform Generator main menu R5-5 thru 6
Waveform manipulation R5-11
Waveform processing and analysis T5-10
Waveform Processor exiting R6-4
Waveform Processor main memory R6-6
Waveform repositories R6-7
Waveform segment processing T5-12
Waveform segments R6-2 thru 3
Waveform types R5-2, R5-7 thru 8
WFM R6-2
Wfrm ovl R6-10
Wfrm window R6-11
Wilcoxon Signed-rank test R11-8
window T2-5, R2-35, R2-30, R4-4
Work files R1-11
Write R7-9
X or Y Parameter R4-7
Xchg R6-9
xsect R2-18, R2-35
xy Auto T2-6, R4-4
xy Axis R4-5
xy Plot R4-4
xy>pol R2-24
xy>z R2-24
xyz>sph R2-25
y Auto T2-6, R4-3
y plot R4-3
Yn R2-26
z>ra R2-24
z>xy R2-24
[]max R2-21
[]min R2-21
[]prod R2-21
[]sum R2-21
^max R2-21
^min R2-21
^sum T7-8, R2-21

